

PETTM Fun and Games

Selected  Programs

Ron Jeffries and
Glen Fisher



PETTM Fun and Games

Selected **CURSORTM** Programs

Ron Jeffries and
Glen Fisher



Disclaimer of Warranties and Limitation of Liabilities

The authors have taken due care in preparing this book and the programs in it, including research, development, and testing to ascertain their effectiveness. The authors and the publishers make no expressed or implied warranty of any kind with regard to these programs nor the supplementary documentation in this book. In no event shall the authors or the publishers be liable for incidental or consequential damages in connection with or arising out of the furnishing, performance, or use of any of these programs.

CURSOR™ is a trademark of The Code Works.

PET and **CBM** are trademarks of Commodore Business Machines, Inc. *PET Fun and Games* is not sponsored or approved by or connected with Commodore Business Machines, Inc. All references to **PET** and **CBM** in the text of this book are to the trademarks of Commodore Business Machines, Inc.

Master Mind® is a registered trademark of Invitica Plastics.

Othello® is a registered trademark of Gabriel Industries Inc.

Yahtzee® is a registered trademark of E.S. Lowe Company Inc.

Chutes and Ladders® is a registered trademark of Milton Bradley Company.

**Published by
OSBORNE/McGraw-Hill
630 Bancroft Way
Berkeley, California 94710
U.S.A.**

For information on translations and book distributors outside of the U.S.A., please write OSBORNE/McGraw-Hill at the above address.

PET™ FUN AND GAMES

Copyright © 1981 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

1234567890 HCHC 87654321

ISBN 0-931988-70-5

Cover art and text illustrations by Jenni Jeffries

Cover concept by KLT van Genderen

ACKNOWLEDGMENTS

We want to thank all of the people who have helped make CURSOR and this book possible. Most important are the authors of the programs. This book is a tribute to their creativity and imagination.

Ralph McElroy deserves special mention, since he is the one who invented the "cassette magazine" idea when he introduced CLOAD for the TRS-80 in early 1978. One of us (Ron) tried to talk Ralph out of what seemed like a crazy idea, but he ignored our advice, and CLOAD became an overnight success story. When we decided to start CURSOR as a cassette magazine for the PET, Ralph was very helpful.

We want to thank all of the people who have subscribed to CURSOR for their support. Obviously, without subscribers, CURSOR wouldn't exist and this anthology of programs would not have been written. But more than that, they've been kind enough to tell us about our mistakes, and always have been a rich source of ideas for programs.

The entire Jeffries family has been part of CURSOR since its inception. Jann Jeffries has kept the books straight, and assisted in almost

every phase of the business. Scott and Jennifer have helped with everything from duplicating tapes to stuffing envelopes.

Bob Albrecht has been a friend and supporter of CURSOR since the early days. Dana Trout, Gregg Pickles, and Stan Reifel have helped us solve all sorts of hardware problems. Nancy Rhodes played a major role in helping prepare this book for publication. She got us organized, contacted program contributors, and supervised much of the program testing.

All of the following people have tested CURSOR programs and deserve credit for finding obscure bugs, and helping to polish the programs: Mike Rowley, Todd Zimmerman, Eric Friesen, Diane Grove, Brian Sawyer, Shawn Meehan, Dave Iwanaga, Andy Stadler, Randi Opp, and Scott Jeffries.

And finally, a special thanks to the great folks at OSBORNE/McGraw-Hill. They've been a delight to work with.

R. J.
G. F.

CONTENTS

ACTION GAMES

ZAP 2

Maneuver your “zapper” around an arena and score points by “zapping” targets.

SHARK 6

Emerging from the depths of the ocean, you are the “jaws” just looking for a meal.

DEMON 13

A demon has escaped during a public demonstration and your job is to capture it.

FERRY 17

Deep in outer space, your mission is to ferry provisions to a stranded expedition.

BAT 21

You are a bat, and you’re very hungry; unfortunately, you have a few handicaps that hinder your eating habits.

NAB 26

It’s off to the races, but the track is littered with money as well as ruthless opponents.

PUZZLES

HANOI 32

If a group of Southeast Asian priests can figure out how to move the golden disks, so can you.

BOX 37

Join the electronics geniuses of Silicon Valley in producing circuit chips for microcomputers.

MIND 43

Try to outwit the PET by guessing the computer’s colors and patterns for a set of pegs.

PEG 49

It’s apparently an easy task to switch two sets of pegs, but a few twists add a bit more challenge.

LEAP 53

Remove pegs one by one simply by leaping over them, but try to end up with just one peg and you’ll discover the difficulty.

FIFTEEN 57

Slide the tiles around the board, but be sure that you finish your juggling with the tiles in numerical order.

RATRUN 60

Now's your chance to wander through a maze in search of cheese, only here it is done from a "rat's eye" view.

GAMES OF RISK

MAZE 70

The PET has constructed a maze filled with gold, but your mission is to find the gold without the aid of a map.

DUNGEON 75

Deep within some dark and dank dungeons, you must outwit some ancient monsters in order to get their gold.

EVEREST 80

Watch your step in this one, as the harsh elements of the world's tallest mountain hinder your trek to the peak.

GODZILLA 85

One of the world's most famous monsters is on the rampage again, and you must defeat it without the help of any giant apes.

FIRE 91

The inferno has started and only you can prevent total disaster, but you must be quick to avoid a pile of smoldering ashes.

GAMES OF STRATEGY

DOTS 98

You may have thought you had played everyone in this favorite game, but you haven't lived until you've played the PET.

REVERSI 103

The game may be as old as time itself, but just try to defeat the PET with strategy and guile.

PIEGRAM 109

Look out because the PET is flinging pies your way, but the catch is that it is invisible while doing so.

COPS 115

It's the old cops and robbers chase routine through the city with some very clever crooks to be chased.

SPOT 120

You only need four squares in a row to win, but the fun is in how you place the squares to get your winning combination.

GAMES OF CHANCE

BJACK 126

Deal another hand because this imaginary card playing game allows you all the money you can spend.

YAHTZEE 133

If you think rolling five dice to come up all with the same number is easy, perhaps you should try this game.

BONZO 140

Play against a friend or against the computer, but this is a race to the top with a few obstacles thrown in.

BOSWAIN 144

Pay attention to this guy's hands because he certainly has a hungry eye on yours.

GAMES FOR FUN

MAD 152

Composing stories may not have been your idea of a good time before, but with the aid of a computer it can be most interesting.

QUIX 156

Remember what you just saw and heard and be able to repeat it exactly, or you'll never defeat the PET.

BOP 160

If you think counting on your fingers is old-fashioned, perhaps you'll reconsider after playing this game.

FACE 167

Your artistic talents can now bubble forth through the screen of the PET as you compose a variety of mugs.

INTRODUCTION

P*ET Fun and Games* provides you with complete listings for 31 playful programs written specifically for the Commodore PET and CBM personal computers. The programs take full advantage of special features of these Commodore computers, including their unique character graphics. All of the programs are written in BASIC, but you don't need to be a programmer to play these games; just enter them from the listings provided.

All of the programs included here have a common goal: to be fun! Some of the programs will test your memory, some will test your strategy, others will challenge your creativity, and still others will offer you some educational value. In some games you may play against another person, but beware, because in others your opponent is the computer.

Some readers may already know of a unique magazine named *CURSOR*. It is unique in part because it publishes programs only for Commodore PET and CBM computers. It is also unique because it publishes the programs on cassette. Subscribers then "read" the magazine by loading the programs into the PETs and CBMs.

The programs included in *PET Fun and Games* have been carefully selected from all the programs published in *CURSOR* over the past three years. Fourteen programmers have contributed to this book, each with individual style and approach. With that, we offer here some of the best, most showy, entertaining and fun programs ever written for the Commodore computers.

How to Use This Book

Complete listings are provided for each program. This means you can type the program in your computer exactly as it is listed here, and then play! The instructions on how to play each game are provided within each program, with supplemental instruction offered in the introduction to each program.

The programs make extensive use of the graphic characters standard to the Commodore PET computers. These graphic characters are part of the standard character set of every PET mode;

you can see the characters on each key on the keyboard. The CBM computer does not normally display graphic characters; instead, it normally displays upper- and lower-case alphabetic characters. If you are using a CBM, you can enable most graphics characters by typing POKE 59468,12 before entering the program.¹ (After entering the program, you can type POKE 59468,14 to return the CBM to its normal upper- and lower-case alphabetic character set.) Some graphics cannot be typed on a CBM with a business keyboard. You will have to enlist the aid of a friend with a PET to enter them.

Whenever you enter any program, be sure to save it on a cassette or diskette so you won't have to type it in again. If you own a CBM but have access to a PET, you can also enter the programs using a PET, and then run them on your CBM.

All of the programs are also available on cassette or diskette from the folks at *CURSOR*. For information, you should write to:

PET FUN AND GAMES
c/o *CURSOR* Magazine
5266 Hollister, Suite 224
Santa Barbara, CA 93111
U. S. A.

How to Read the Listings in This Book

We have used special notation in the listings in order to make them easier to read. You should become familiar with this notation before you begin to enter any listing.

First and foremost, anything printed within curly braces is special notation. Curly braces look like this: { }. Anything printed within these braces, including the braces themselves, should not be keyed into your listing as you see it. All special notation is explained in the following paragraphs.

You should, however, *enter every character not printed within a set of curly braces exactly as you see it printed in the listing, with one exception: spaces.*

¹ For more information on substituting graphic characters for alphabetic on a CBM, consult *PET/CBM Personal Computer Guide*, second edition, by Adam Osborne and Carroll S. Donahue (OSBORNE/McGraw-Hill, 1980).

Because it is sometimes hard to “see” a space character within a listing, we use an underlined period (it looks like this: .) to represent a space character which must be entered in an exact location or sequence; this usually occurs within a PRINT statement. Normally, we do not use a . character for a space which appears merely as a separator of words, but you should also enter these spaces for easier reading of your listings (unless the space is between words within curly braces, illustrated below). For now, you must remember to *enter a space character whenever you see the . character*. For example, the PRINT statement

```
PRINT "GREETINGS, EARTHLING! TAKE US TO YOUR LEADER!"
```

is shown as

```
PRINT "GREETINGS, .EARTHLING! .TAKE .US .TO .YOUR .LEADER!"
```

There are two kinds of special characters within curly braces: graphics characters (shift-whatever), and cursor control characters (for instance, cursor down, or home cursor). Graphics characters are shown within braces as their unshifted counterparts. Thus, the solid square would be shown as “{&}”, the diamond would be “{Z}”, and a shifted space would be shown as “{.}”.

Cursor control characters are shown within braces as the name of the key. The name of any cursor control key is printed in lower-case letters to avoid confusing the name with an odd sequence of graphics characters.

{up}	Cursor Up
{down}	Cursor Down
{left}	Cursor Left
{right}	Cursor Right
{rvs}	Reverse Video
{off}	Normal Video (reverse off)
{home}	Home Cursor
{clr}	Clear Screen
{ins}	Insert Character
{del}	Delete Character

Quite often, cursor control keys are repeated many times. For example, it's common to program a sequence like “{home down down down down down down}”. To save space in the printed listings, and to simplify your work when typing the programs, we tell you how many times to repeat a cursor control key. The “press-count” comes right before the key name, separated by a degree (°) character. Printed that way, the above example is listed out as “{home 7°down}”.

There is one catch to the “press-count” notation. There are a few graphics that get printed

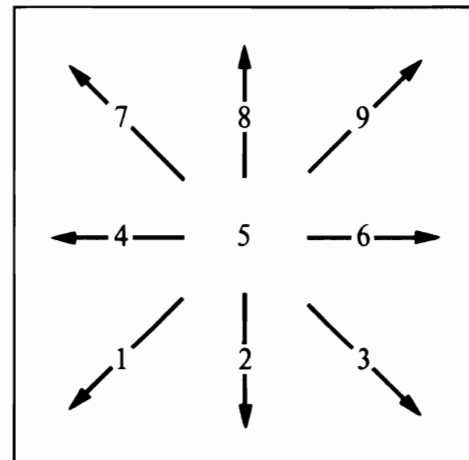
as numbers within braces. How can you tell when a digit inside braces represents a graphics character, and when it means a “press-count”? Since those particular graphics are not used much, a number within braces will most likely be a “press-count”, but the sure way to tell is to see if it is followed by a “°” and a cursor control. If so, it is a press-count. If it is followed by *anything* but a “°”, it is a graphics character. For example, when you see this line in a listing:

```
20 PRINT "{02. 3°left down}"
```

you enter: [2] [0] [space] [P] [R] [I] [N] [T]
[space] ['] ['] [shift-0] [shift-2] [shift-period]
[cursor left] [cursor left] [cursor left] [cursor
down] ['] ['].

Numeric Keypad

The numeric keypad is used to control movement in many of the games. As shown in the diagram below, you control movement on the screen by pressing the numeric key that has that position on the pad.



For example, to move an object up (or “forward”), press [8]. To move right, press [6], while [4] moves you left. To move down, press [2]. Some games also allow diagonal movement, using the [1], [3], [7] and [9] keys.

Sound

Several of the games use the CB2 sound capability of the PET. These games are much more fun if you have sound, so it is strongly suggested that you purchase a sound attachment. (If you have one of the new “Fat Forty” models, a sound device is built-in.) If your local

Commodore dealer doesn't have a sound adapter, write to Channel Data, 4141 State, Suite A2, Santa Barbara, CA, or to CAP Electronics, 8462 Hillwood Lane, Tucson, AZ 85715.

Another note about sound is that "CB2" is produced by using the 6522 chip, which also produces the tones that are used to record programs on the cassette recorder. If you use the Stop key to get out of a program that uses sound, you may find that you can't save a program. You can correct this program by typing the following line, which contains three commands which

restore the 6522 sound registers to their normal condition:

POKE 59464,0: POKE 59466,0: POKE 59467,0

Games that use sound are denoted in each program heading by the following symbol:





ACTION GAMES



ZAP

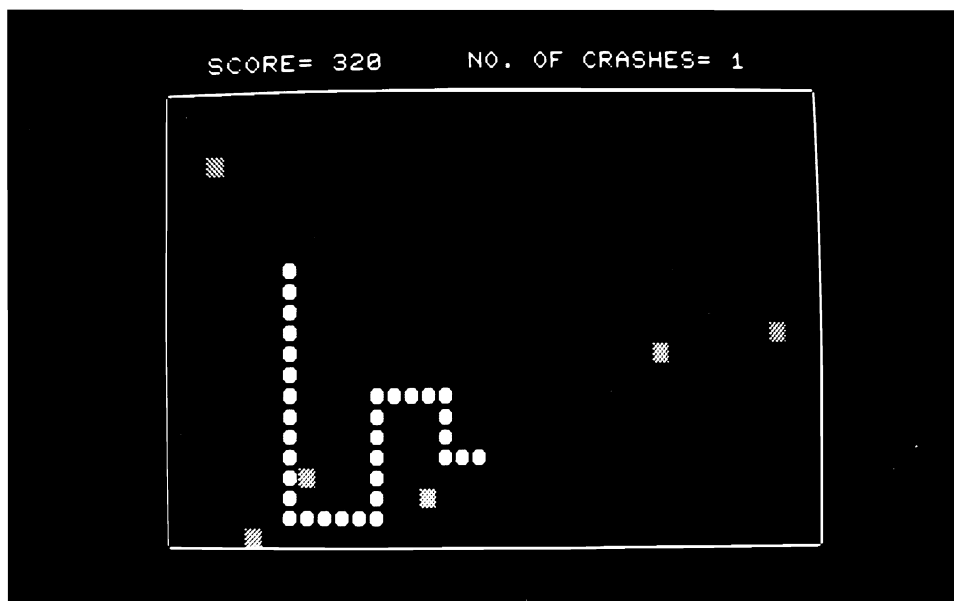
By Tom Marazita

When the game begins, a snake-like train of dots, (your zapper) starts crawling up the right-hand edge of the arena. Using your zapper, try to "zap" all the targets that pop up on your screen. The hitch is that you must do this without driving your "zapper" either into a wall or back across itself. After you get used to controlling your zapper, it will not be hard to avoid hitting a wall. However, each time you zap a target, your zapper becomes longer, and it becomes increasingly difficult to maneuver without running into yourself.

You control the zapper using the numeric

keypad. Each time you run over a target you get ten points. Pressing [5] allows you to stop where you are, but while you are stopped, your score automatically decreases very rapidly. If you spend a lot of time wandering about the arena but not zapping much of anything, many targets will appear and then disappear unzapped, thus reducing your final percentage score.

The game is over after you have crashed six times. The PET will tell you what your score was, what the maximum score for that session would have been if you had zapped all of the targets, and what percentage of the targets you actually did zap.



```
0 CLR:POKE 59468,12
6 REM ZAP COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="ZAP":NM$="2":GOSUB 62000
100 DIM WM(255),A(255)
101 GOSUB 60400
102 EL=18:HP=EL-1
120 PRINT "{clr}"
130 PO=CRT+WD*25-2
140 S=1
160 FOR X=CRT+WD+1 TO CRT+WD*2-2:POKE X,100:NEXT
170 FOR X=CRT+WD*2 TO CRT+(WD*23) STEP WD:POKE X,103:NEXT
180 FOR X=CRT+(WD*24)+1 TO CRT+WD*25-2:POKE X,99:NEXT
```



```

190 FOR X=CRT+WD*24-1 TO CRT+WD*3-1 STEP -WD:POKE X,101:NEXT
195 GOSUB 900
200 GET A$:IF A$<>" " THEN 200
211 GOTO 410
220 A$="8":GOTO 240
230 GET A$:IF A$="Q" THEN 650
240 IF A$="4" THEN DIR=-1
250 IF A$="8" THEN DIR=-WD
260 IF A$="6" THEN DIR=1
270 IF A$="2" THEN DIR=WD
280 IF A$="7" THEN DIR=-WD-1
290 IF A$="9" THEN DIR=-WD+1
300 IF A$="3" THEN DIR=WD+1
310 IF A$="1" THEN DIR=WD-1
311 IF A$="5" THEN 770
320 PO=PO+DIR
321 IF PO=33767 OR PO=33728 OR PO=32808 OR PO=32847 THEN 590
324 IF PEEK(PO)=102 THEN GOSUB 550
325 WM(HP)=PO:HP=HP+1
326 IF HP>EL THEN HP=0
340 IF PEEK(PO)>98 AND PEEK(PO)<102 THEN 590
345 IF PEEK(PO)>102 AND PEEK(PO)<104 THEN 590
350 IF PEEK(PO)=81 THEN 590
360 POKE PO,81
370 R=INT(RND(1)*15)+1:IF S>5 THEN ZZ=20
375 P1=INT(RND(1)*ZZ)+1:ZZ=50
380 IF R=3 THEN 410
390 IF P1=3 THEN 480
400 GOSUB 730:GOTO 230
410 REM
420 IF S>6 THEN 400
430 Q=INT(RND(1)*1000)+(32768+160)
435 IF Q>33726 THEN 430
440 IF PEEK(Q)<>32 THEN 430
450 S=S+1:H=H+1:TC=TC+1
460 POKE Q,102
470 A(H)=Q:IF CV=1 THEN 400
475 CV=1:GOTO 220
480 REM
490 P=INT(RND(1)*H)+1
500 REM
510 IF PEEK(A(P))<>102 THEN 540
520 POKE A(P),32
530 S=S-1
540 A(P)=A(H):H=H-1:GOTO 400
550 REM
560 SC=SC+10
565 S=S-1
566 EL=EL+2
567 IF EL>250 THEN EL=250
570 GOSUB 900

```

```

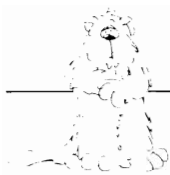
580 RETURN
590 CA=CA+1
595 GOSUB 900
600 IF CS>L THEN L=CS
610 CS=0
615 EL=18:HP=17
616 TP=0
620 CV=0
635 FOR T=1 TO 8:FOR X=1 TO 100:NEXT:POKE PO,209:FOR X1=1 TO 100:
    NEXT:POKE PO,81:NEXT
636 FOR X=0 TO 255:WM(X)=0:NEXT:HP=17:TP=0
638 IF CA=6 THEN 650
640 GOTO 120
650 PRINT "{clr 6°down}"
660 PRINT "YOUR..SCORE..IS"SC
662 PRINT
666 PRINT
667 PRINT "HIGHEST..POSSIBLE..SCORE:"TC*10
670 PRINT
675 PRINT "YOU..HIT";INT((SC/(TC*10))*100);"%..OF..THE..TARGETS"
677 GET Q$:IF Q$<>" " THEN 677:REM WAS POKE525,0
680 PRINT "{down}WANT..TO..PLAY..AGAIN?"
681 GET RJ$:IF RJ$="" THEN 681
682 IF RJ$<>"N" THEN CLR:GOTO 100
685 PRINT "{down}HOPE..YOU..HAD..FUN!{down}"
700 END
730 LOC=WM(TP):TP=TP+1
740 IF TP>EL THEN TP=0
750 POKE LOC,32
760 RETURN
770 FOR E=1 TO 171:GET A$:IF A$<>" " THEN 240
780 NEXT E:SC=SC-10:IF SC<0 THEN SC=0
790 GOSUB 900
800 GOTO 770
900 PRINT "{home 3°right}SCORE="SC;TAB(18);"NO..OF..CRASHES="CA
910 RETURN
60000 IN$="..":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "._{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$,:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000

```

```

60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTIUNE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_{-}_CURSOR_{#}";NM$;
      "_{(C)}_{1981}_{PAGE}";PN;"{down}"
60270 GOSUB 60500:RETURN
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"_{BY}_TOM_{MARAZITA}"
62010 PRINT "{down}_{CURSOR}_{#2}_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62025 PRINT "{down}CHASE_{THE}_TARGETS"
62030 PRINT "{2°down}WANT_{DIRECTIONS?}_{}";:GOSUB 60000
62040 IF IN$<>"Y" THEN 100
62100 LP=1:GOSUB 60200
62130 PRINT "YOU_{CONTROL}_{A}_STRANGE_{MISSILE}!"
62140 PRINT "{down}TARGETS_{'{'}}_{APPEAR}_{ON}_{THE}_{SCREEN}_{,}_{WHICH}"
62150 PRINT "YOU_{TRY}_{TO}_{HIT}_{...}_{USE}_{NUMBERS}_{TO}_{CONTROL}"
62160 PRINT "THE_{MISSILE}_{:}_{8}_{MOVES}_{YOU}_{UP}_{,"
62170 PRINT "_{...}_{2}_{MOVES}_{YOU}_{DOWN}_{."
62180 PRINT
62190 PRINT "_{...}_{4}_{MOVES}_{YOU}_{LEFT}"
62200 PRINT "_{...}_{6}_{MOVES}_{YOU}_{RIGHT}_{."
62202 PRINT
62205 PRINT "(1,_{3},_{7},_{AND}_{9}_{MOVE}_{YOU}_{DIAGONALLY})"
62210 PRINT
62212 PRINT "EACH_{TARGET}_{IS}_{WORTH}_{10}_{POINTS}_{,}_{BUT}"
62225 PRINT "YOUR_{MISSILE}_{GETS}_{BIGGER}_{AFTER}_{HITTING}"
62228 PRINT "A_{TARGET}_{...}"
62230 PRINT "{down}THE_{GAME}_{ENDS}_{AFTER}_{6}_{CRASHES}!"
62235 GOSUB 60200

```



SHARK

By Glen Fisher

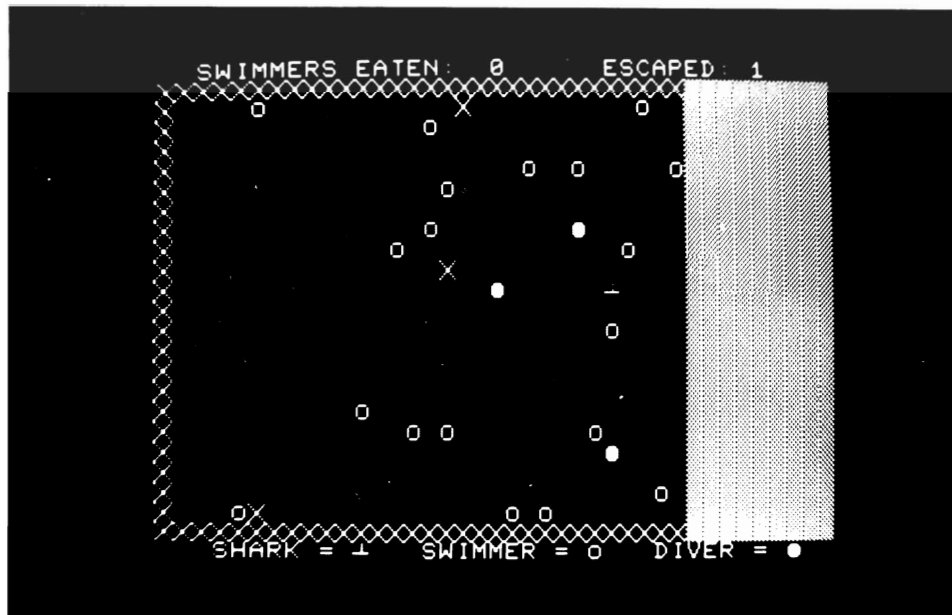
It's time for dinner, and we have just the thing — man-eating shark. Or is it shark, eating man? In this tasty little program, your task is to munch as many swimmers as you can. But beware of the divers and traps; they both make life difficult for you. Also, the swimmers are not at all cooperative — they're busy heading for the beach, trying to escape your jaws.

The PET will start by asking how many swimmers, how many traps, and how many divers you want. Make things as difficult or easy as you like (there are some limits). The swimming area is surrounded by shark traps on three sides and the beach on the fourth. Since "they" know you're there, "they've" sent out divers and put more shark traps into the swimming area, all in an effort to get you before you get them.

When your shark symbol ($\perp$) is flashing, it's

your turn to move (which is done using the numeric key pad). Being a shark, you're a pretty good swimmer, so you normally move two spaces. If you want to move only one space (to avoid a trap, perhaps), hold [SHIFT] down while you press the number key. After you move, each of the swimmers and divers gets to move one space.

To eat a swimmer, just attack him. Eating swimmers doesn't prevent you from moving — you may attack a swimmer next to you, and continue on to the space beyond (which may hold *another* swimmer). The hazards to avoid are traps, divers, and the beach itself. The divers are after you with spear guns. If they catch you, you're dead. If you attack a diver, you have a 50% chance of winning. Dinner is over when all the swimmers have either hit the beach or been eaten, or when you've been killed by a diver or a trap.



```
0 CLR:POKE 59468,12
6 REM SHARK (C) 1981 THE CODE WORKS
10 REM AS OF 10/06/81 RJ
90 PG$="SHARK":NM$="1":GOSUB 62000
100 GOSUB 60400:HT=25:SF=30:BT=(HT-2)*WD:RT=10
105 BL=32:SH=113:DV=81:SW=87:TR=86:BE=102
```

```

110 PRINT "{down}HOW.MANY.TRAPS?.";GOSUB 60000
115 IF IN$="" THEN IN$="5":PRINT "HOW.ABOUT.";IN$;".TRAPS, THEN?"
120 NT=VAL(IN$):IF NT>25 THEN
    PRINT "TRYING.TO.MAKE.SWIMMING.DANGEROUS?":GOTO 110
125 IF NT<1 THEN PRINT "THAT'S.A.CHALLENGE?":GOTO 110
130 PRINT "{down}HOW.MANY.SWIMMERS?.";GOSUB 60000:NS=VAL(IN$)
135 IF IN$="" THEN IN$="25":PRINT "OK, YOU'LL.GET.";IN$;".SWIMMERS."
137 NS=VAL(IN$)
140 IF NS>100 THEN PRINT "HEALTH.REGULATIONS.PROHIBIT.THAT.MANY.":
    GOTO 130
145 IF NS<1 THEN PRINT "ON.A.STARVATION.DIET?":GOTO 130
150 PRINT "{down}HOW.MANY.DIVERS?.";GOSUB 60000:ND=VAL(IN$)
155 IF IN$="" THEN IN$="2":PRINT "LET.THERE.BE.";IN$;".DIVERS."
160 ND=VAL(IN$):IF ND>25 THEN PRINT "DIVING.ISN'T.THAT.POPULAR!":
    GOTO 150
165 IF ND<1 THEN PRINT "SHARKS.AREN'T.AN.ENDANGERED.SPECIES":
    GOTO 150
185 ES=0:KD=0
186 DEF FN R(N)=INT((N-1)/3)-1:DEF FN C(N)=N-3*INT((N-1)/3)-2
187 DIM SS(NS,1),DS(ND,1)
188 FOR I=0 TO 500:NEXT I
189 PRINT "{clr}...SWIMMERS.EATEN:.....ESCAPED:"
190 FOR I=0 TO SF
191 POKE CRT+WD+I,TR:POKE CRT+I+BT,TR
192 NEXT I
193 FOR I=WD TO BT STEP WD
194 POKE CRT+I,TR
195 FOR J=SF+1 TO 39
196 POKE CRT+I+J,BE
197 NEXT J
198 NEXT I
199 FOR I=1 TO NT
200 GOSUB 10000
210 POKE LOC,TR
220 NEXT I
230 FOR I=1 TO NS
240 GOSUB 10000
245 SS(I,0)=ROW:SS(I,1)=COL
250 POKE LOC,SW
260 NEXT I
270 FOR I=1 TO ND
280 GOSUB 10000
285 DS(I,0)=ROW:DS(I,1)=COL
290 POKE LOC,DV
300 NEXT I
310 GOSUB 10000:POKE LOC,SH
320 SR=ROW:SC=COL
325 PRINT "{home
    24°down}...SHARK.=.{1}...SWIMMER.=.{W}...DIVER.=.{Q}";
330 PRINT "{home}":PRINT "{up}";TAB(18);KD;TAB(34);ES
500 GOSUB 10100

```

```

520 POKE OLD,BL
530 FOR I=1 TO DIST
540 OLD=OLD+CC+WD*RC
545 SC=SC+CC:SR=SR+RC
550 IF PEEK(OLD)=SW THEN GOSUB 10300:POKE OLD,BL:GOTO 570
560 IF PEEK(OLD)<>BL THEN GOTO 5000
570 NEXT I
580 POKE OLD,SH
590 IF ND<1 THEN 790
600 FOR I=1 TO ND
610 RM=HT+1:CM=WD+1
620 ROW=DS(I,0):COL=DS(I,1)
625 OLD=CRT+WD*ROW+COL
626 POKE OLD,BL
630 RC=SGN(SR-ROW):IF RC=0 THEN RC=1
631 CC=SGN(SC-COL):IF CC=0 THEN CC=1
635 LOC=OLD+RC*WD+CC
640 FOR J=RC TO -RC STEP -RC
650 FOR K=CC TO -CC STEP -CC
660 CH=PEEK(LOC):IF CH=BL OR CH=SH THEN 720
680 LOC=LOC-CC
690 NEXT K
700 LOC=LOC+3*CC-WD*RC
710 NEXT J
715 BUG=1:GOTO 50000
720 ROW=ROW+J:COL=COL+K
730 LOC=CRT+WD*ROW+COL
740 DS(I,0)=ROW:DS(I,1)=COL
750 CH=PEEK(LOC):POKE OLD,BL:POKE LOC,DV
760 IF CH=SH THEN CAUSE=1:GOTO 5100
765 IF CH<>BL THEN BUG=2:GOTO 50000
770 NEXT I
790 CT=NS
800 FOR I=1 TO NS
810 ROW=SS(I,0):COL=SS(I,1)
820 RC=SGN(ROW-SR):IF RC=0 THEN RC=SGN((HT-ROW)-ROW):IF RC=0 THEN
    RC=1
830 OLD=CRT+WD*ROW+COL
840 LOC=OLD+WD*RC+1
850 FOR J=RC TO -RC STEP -RC
860 FOR K=1 TO -1 STEP -1
865 CH=PEEK(LOC)
870 IF ABS(ROW+J-SR)+ABS(COL+K-SC)>1 AND (CH=BL OR CH=BE) THEN 930
880 LOC=LOC-1
890 NEXT K
900 LOC=LOC+3-WD*RC
910 NEXT J
920 J=0:K=0:LOC=OLD
930 ROW=ROW+J:COL=COL+K
935 POKE OLD,BL
940 IF PEEK(LOC)=BE THEN GOSUB 10500:GOTO 960

```

```

945 POKE LOC,SW
950 SS(I,0)=ROW:SS(I,1)=COL
960 NEXT I
970 IF CT=NS THEN 500
980 FOR I=1 TO CT
990 IF SS(NS,0)<0 THEN NS=NS-1:GOTO 990
1000 IF SS(I,0)<0 THEN SS(I,0)=SS(NS,0):SS(I,1)=SS(NS,1):NS=NS-1
1010 NEXT I
1020 GOTO 500
5000 CH=PEEK(OLD)
5010 IF CH=DV AND RND(1)<.5 THEN GOSUB 10300:GOTO 570
5030 IF CH=DV THEN CAUSE=1:GOTO 5100
5040 IF CH=BE THEN POKE OLD,SH+128:CAUSE=2:GOTO 5100
5050 IF CH=TR THEN CAUSE=3:GOTO 5100
5090 BUG=3:GOTO 50000
5100 FOR I=1 TO CAUSE:READ M$:NEXT I
5120 PRINT "{home 23°down}";
5130 PRINT ".....";CHR$(20);
    "{up}"
5140 PRINT TAB((30-LEN(M$))/2);"{rvs VVV off}";M$;"!!{rvs VVV
    off}"
5210 PRINT "DO YOU WANT TO PLAY AGAIN?.....{up}"
5215 PRINT TAB(27);:GOSUB 60000
5220 IF LEFT$(IN$,1)<>"N" THEN CLR:GOTO 100
5250 PRINT "{clr 6°down}THANKS FOR PLAYING."
5260 END
5500 DATA "THE DIVER GOT YOU"
5510 DATA "YOU BEACHED YOURSELF"
5520 DATA "YOU GOT TRAPPED"
5530 DATA "YOU GOT THE LAST SWIMMER"
5540 DATA "THE LAST SWIMMER JUST ESCAPED"
6010 END
10000 ROW=INT(RND(1)*(HT-3))+2:COL=INT(RND(1)*SF)+1
10010 LOC=CRT+WD*ROW+COL:IF PEEK(LOC)<>BL THEN 10000
10017 IF LOC>CRT+WD*25-1 THEN 10000
10020 RETURN
10100 OLD=CRT+WD*SR+SC:TM=TI:DK=-1
10105 GET C$:IF C$<>" " THEN 10105
10110 GET C$:IF C$<>" " THEN 10160
10120 IF TI<TM THEN 10110
10130 IF DK THEN POKE OLD,BL
10140 IF NOT DK THEN POKE OLD,SH
10150 DK=NOT DK:TM=RT+TI:GOTO 10110
10160 TE=(ASC(C$) AND 127)-ASC("0")
10165 IF C$="Q" THEN 5250
10170 IF TE<1 OR TE>9 THEN 10110
10180 DIST=1:RC=FN R(10-TE):CC=FN C(TE)
10190 IF ASC(C$)<128 THEN DIST=2
10200 RETURN
10250 TM=TI+DL
10260 IF TI<TM THEN 10260

```

```

10270 RETURN
10300 CH=PEEK(OLD):DL=5
10305 POKE OLD,SH
10310 FOR II=1 TO 4
10320 GOSUB 10250:POKE OLD,CH
10330 GOSUB 10250:POKE OLD,SH
10340 NEXT II
10350 POKE OLD,BL
10360 IF CH=SW THEN 10420
10370 FOR II=1 TO ND
10380 IF DS(II,0)=SR AND DS(II,1)=SC THEN 10400
10390 NEXT II:BUG=4:GOTO 50000
10400 DS(II,0)=DS(ND,0):DS(II,1)=DS(ND,1):ND=ND-1
10410 RETURN
10420 FOR II=1 TO NS
10430 IF SS(II,0)=SR AND SS(II,1)=SC THEN 10450
10440 NEXT II:BUG=5:GOTO 50000
10450 SS(II,0)=SS(NS,0):SS(II,1)=SS(NS,1):NS=NS-1:KD=KD+1
10455 PRINT "{home}";TAB(18);STR$(KD);".."
10460 IF NS>0 THEN RETURN
10470 POKE OLD,SH:CAUSE=4:GOTO 5100
10500 DL=1:LOC=CRT+WD*ROW+(SF+1)
10510 FOR II=SF+1 TO 39
10520 POKE LOC,SW+128:GOSUB 10250
10530 POKE LOC,BE:LOC=LOC+1
10540 NEXT II
10550 SS(I,0)=-1
10560 CT=CT-1:ES=ES+1
10565 PRINT "{home}";TAB(34);STR$(ES);".."
10570 IF CT>0 THEN RETURN
10580 CAUSE=5:GOTO 5100
20000 DL=20
20010 GOSUB 10250:POKE LOC,CH
20020 GOSUB 10250:POKE LOC,DV
20030 GOTO 20010
20100 DL=16:CH=PEEK(OLD)
20110 GOSUB 10250:POKE OLD,(CH+128) AND 255
20120 GOSUB 10250:POKE OLD,CH
20130 GOTO 20110
50000 T$=STR$(BUG):T=LEN(T$)+3
50010 PRINT "{home , off}";
50020 FOR IJ=1 TO T:PRINT "{"";:NEXT IJ
50030 PRINT "{,}":PRINT "{rvs !}BUG";T$;"{off !}"
50040 PRINT "{< rvs}";
50050 FOR IJ=1 TO T:PRINT "{"";:NEXT IJ
50060 PRINT "{off >}"
60000 IN$="..":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010

```



```

60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>10 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_{CURSOR}_{#}";NM$;
      "_{(C)}_{1981}_{PAGE}";PN;"{down}"
60270 GOSUB 60500:RETURN
60400 CRT=32768:CR$=CHR$(13):PRINT "{clr}"
60410 WD=40:POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:
      PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT "{left ins D}":RETURN
62000 PRINT "{clr 2°down}";TAB(9);PG$;"_{BY}_{GLEN}_{FISHER}"
62010 PRINT "{down}_{CURSOR}_{#1}_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62030 PRINT "{down}EAT_{ALL}_{THE}_{SWIMMERS}_{BEFORE}_{THEY}_{ESCAE}{left
      ins}P"
62040 PRINT "{2°down}WANT_{DIRECTIONS?}_{}";:GOSUB 60000
62050 IF IN$<>"Y" THEN 100
62100 LP=2:GOSUB 60200
62130 PRINT "YOU_{ARE}_{A}_{SHARK}_{'1}'"
62140 PRINT "THAT_{EATS}_{SWIMMERS}_{'W}'"
62150 PRINT "WHILE_{AVOIDING}_{DIVERS}_{'Q}'"
62160 PRINT "AS_{WELL}_{AS}_{TRAPS}_{'V}'"
62170 PRINT
62180 PRINT "USE_{THE}_{NUMBER}_{KEYS}_{TO}_{MOVE}."
62185 PRINT ""
62190 PRINT "THE_{'5'}_{KEY}_{IS}_{YOUR}_{CURRENT}_{LOCATION}."
62195 GOSUB 60200
62200 PRINT "FOR_{EXAMPLE}_{,}_{TO}_{MOVE}_{TWO}_{SPACES}"
62210 PRINT "TO_{THE}_{RIGHT}_{,}_{PRESS}_{THE}_{'6'}_{KEY}"
62220 PRINT
62230 PRINT "_{7}_{8}_{9}"
62240 PRINT "_{M}_{B}_{N}"
62250 PRINT ""
62260 PRINT "_{4}{C}_{5}_{C}6"
62270 PRINT ""

```

```
62280 PRINT ".....{N}..{B}..{M}"
62290 PRINT ".....1..2..3"
62300 PRINT
62310 PRINT "SHIFTING..A..NUMBER..MOVES..ONE..SPACE"
62320 PRINT "IN..THAT..DIRECTION..(INSTEAD..OF..TWO)"
62325 GOSUB 60200
```



DEMON

By Ken Morley



Your task is to recapture a demon that has accidentally escaped during the demonstration of a spell. As long as the demon is avoiding capture, he can't free any more demons. The problem is that, as long as the spell is active, another demon can escape once

the first one is recaptured. Thus, you've got to keep after demons until the spell wears off. When the spell expires, the first escaped demon will be sent back where he belongs, but any demons he has freed will remain. It is, therefore, vital that he not be given the chance to free any other demons.

```
0 CLR:POKE 59468,12
6 REM DEMON! COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="DEMON!":NM$="11":GOSUB 62000
100 REM
110 FOR I=1 TO 9:READ N2(I):NEXT I
120 DATA 17,14,13,8,0,12,9,10,11
130 FOR I=1 TO 8:D(I)=I-(I>4):NEXT I
140 T=WD:FOR I=2 TO 9 STEP 3:FOR J=-1 TO 1:E(I+J)=T+J:NEXT:T=T-WD:
    NEXT I
160 FOR I=1 TO 9:READ N1(I):NEXT
170 DATA 58,52,46,28,0,43,30,34,38
190 FQ=59464:TN=59466:CT=59467
300 NM=0:CR$=CHR$(13)
310 OI=33708:TI$="000000"
320 PRINT "{clr}"
480 GOTO 500
490 PRINT "{down}PLEASE ANSWER '1','2',OR '3'."
500 PRINT "{down}WANT A (1) SHORT, (2) MEDIUM,OR"
510 PRINT "(3) LONG GAME?";GOSUB 60000
520 IF IN$="" THEN PRINT "I'LL ASSUME A SHORT GAME.":FOR T=1 TO
    1000:NEXT:IN$="1"
530 IF LEN(IN$)>1 THEN 490
540 IF IN$<"1" OR IN$>"3" THEN 490
550 LM=3600*VAL(IN$)
800 GOSUB 10000
810 FOR I=1 TO 5:POKE OI,81:FOR J=1 TO 100:NEXT:POKE OI,32:
    FOR J=1 TO 100:NEXT:NEXT I
820 POKE OI,81
900 OP=32809+INT(RND(1)*918)
910 IF PEEK(OP)<>32 THEN 900
1000 ND=8
1005 T=INT(RND(1)*ND):DN=D(T):D(T)=D(ND):D(ND)=DN:ND=ND-1
1006 IF PEEK(OP+E(DN))<>32 THEN 1005
1010 FOR I=1 TO INT(RND(1)*6)+3
1020 NP=OP+E(DN)
```

```

1030 IF PEEK(NP)=81 THEN GOSUB 5000:GOTO 900
1040 IF PEEK(NP)=32 THEN POKE OP,32:POKE NP,87:OP=NP:PC=N2(DN):
      GOSUB 4000
1100 IF TI>LM THEN GOTO 2000
1105 POKE QP,255:GET B$:IF B$="." THEN 2000
1110 B=5:IF B$<>" " THEN B=VAL(B$):IF B=0 THEN B=5
1120 N1=O1+E(B)
1130 IF PEEK(N1)=87 THEN GOSUB 5000:GOTO 900
1140 IF PEEK(N1)=32 THEN POKE O1,32:POKE N1,81:O1=N1:PC=N1(B):
      GOSUB 4000
1500 NEXT I
1510 GOTO 1000
2000 PC=0:GOSUB 4000:POKE TN,1:PC=255
2001 POKE OP,102:GOSUB 2500
2002 POKE OP,224:GOSUB 2500
2003 POKE OP,32:GOSUB 2500
2004 GOSUB 4100
2010 FOR I=1 TO 1000:NEXT I
2015 PRINT "{clr 3°down}THE SPELL HAS EXPIRED!"
2020 PRINT "{down}YOU CAUGHT";NM;"DEMON";
2025 IF NM<>1 THEN PRINT "S";
2030 PRINT "!":FOR I=1 TO 100:NEXT I
2035 GET T$:IF T$<>" " THEN 2035
2040 PRINT "{down}WANT TO PLAY AGAIN?";:GOSUB 60000
2050 IF LEFT$(IN$,1)<>"N" THEN 300
2100 PRINT "{clr 2°down}":END
2500 FOR I=1 TO 20:FOR J=1 TO 10:NEXT J:POKE FQ,PC:PC=PC-4:NEXT I:
      RETURN
4000 POKE CT,16:POKE TN,15:POKE FQ,PC:IF PC=0 THEN RETURN
4100 POKE FQ,0:POKE TN,0:POKE CT,0:RETURN
5000 POKE OP,215
5010 PC=0:GOSUB 4000
5020 FOR K=1 TO 250 STEP 2:POKE FQ,K:NEXT
5030 GOSUB 4100
5040 NM=NM+1
5050 RETURN
10000 T$="":FOR I=1 TO WD-3:T$=T$+"{@" :NEXT I
10010 PRINT "{clr 0}";T$;"{. left ins @"
10020 FOR I=1 TO 23:PRINT "{|}";TAB(WD-2);"{| left ins}":NEXT I
10030 PRINT "{-}";T$;"{= left ins @ home}"
10035 FOR K=1 TO 70
10040 W=INT(RND(1)*(WD*23-2)+WD+1)
10050 IF PEEK(32768+W)<>32 THEN 10040
10060 POKE 32768+W,214
10070 NEXT
10080 RETURN
60000 IN$="." :ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010

```

```

60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{to}";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"{-}_CURSOR_{#}";NM$;
      "{_}(C)_{1981}_{_}PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}";:GOSUB 60400:GOTO 100
60400 CRT=32768:WD=40:CR$=CHR$(13):PRINT "{clr}":QP=515:
      IF PEEK(50000) THEN QP=151
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "@@@";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"{_}BY_{_}KEN_{_}MORLEY"
62005 PRINT "{down}_{_}_{_}_{_}_{_}_{_}CURSOR_{_}#11_{_}COPYRIGHT_{_}(C)_{_}1981"
62010 GOSUB 60500
62030 PRINT "{down}CAPTURE_{_}THE_{_}ESCAPED_{_}DEMONS"
62040 PRINT "{2°down}WANT_{_}DIRECTIONS?_{_}";:GOSUB 60000
62050 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=3:GOSUB 60200
62110 PRINT "IN_{_}THE_{_}MIDST_{_}OF_{_}A_{_}DEMONSTRATION_{_}OF_{_}A_{_}NW{left ins}E"
62120 PRINT "{down}SPELL_{_}AT_{_}THE_{_}WIZARDS_{_},_{_}WARLOCKS_{_}AND_{_}SOR-"
62130 PRINT "{down}CERERS_{_}CHOWDER_{_}AND_{_}MARCHING_{_}SOCIETY_{_},_{_}AN"
62140 PRINT "{down}ACCIDENT_{_}OCCURRED_{_},_{_}PERMITTING_{_}THE_{_}ESCAE{left
      ins}P"
62150 PRINT "{down}OF_{_}SOME_{_}DEMONS_{_}._{_}FORTUNATELY_{_},_{_}THE_{_}SPELL"
62155 PRINT "{down}WAS_{_}SET_{_}UP_{_}SO_{_}THAT_{_}IT_{_}WILL_{_}EXPIRE_{_}SOON."
62160 PRINT "{down}ALSO_{_},_{_}ONLY_{_}ONE_{_}DEMON_{_}CAN_{_}BE_{_}LOOSE_{_}AT_{_}A"
62170 PRINT "{down}TIME_{_}(WHICH_{_}IS_{_}JUST_{_}AS_{_}WELL_{_}...)_{_}._{_}YOUR"
62180 PRINT "{down}TASK_{_}IS_{_}TO_{_}RECAPTURE_{_}AS_{_}MANY_{_}DEMONS_{_},_{_}AS"
62190 PRINT "{down}POSSIBLE_{_}BEFORE_{_}THE_{_}SPELL_{_}EXPIRES."
62200 GOSUB 60200
62201 PRINT "THE_{_}DEMONS_{_}MUST{4°left down ###
      up}_{_}BE_{_}RECAPTURED_{_};_{_}IF_{_}LEFT"
62202 PRINT "{down}FREE_{_},_{_}THEY_{_}WILL_{_}TRY_{_}TO_{_}FREE_{_}STILL_{_}MORE"

```

```

62203 PRINT "{down}DEMONS..THAT.CANNOT{6°left down #####
        up}.BE.ALLOWED!"
62210 PRINT "{2°down}YOU.MAY.CONTROL.YOUR.MOVES.WITH.THE"
62220 PRINT "{down}NUMERIC.PAD..THE.NUMBER.PAD.....7.8.9"
62230 PRINT TAB(35);"{M|N}"
62240 PRINT "MOVES.YOU.AS.SHOWN.TO.THE.RIGHT...4{@Q@}6"
62250 PRINT TAB(35);"{N|M}"
62260 PRINT "TO.STOP.THE.GAME.EARLY,.PRESS.....1.2.3"
62270 PRINT "{down}THE.{rvs}SPACE{off}.KEY."
62310 GOSUB 60200
62320 PRINT "IN.THE.GAME,.YOU.WILL.BE.THE.'{Q}',.AND"
62330 PRINT "{down}THE.CURRENTLY.ESCAPED.DEMON.WILL.BE.'{W}'.{left
        ins}'"
62340 PRINT "{down}WHEN.YOU.CAPTURE.A.DEMON,.IT.IS.FROZEN"
62350 PRINT "{down}IN.PLACE,.AND.MARKED.WITH.'{rvs W off}'.WHEN.THE"
62360 PRINT "{down}SPELL.ENDS,.THE.CURRENTLY.ESCAPED.DEMON"
62370 PRINT "{down}WILL.BE.SENT.BACK.WHERE.HE.CAME.FROM,"
62380 PRINT "{down}AND.WON'T.BE.A.PROBLEM..ANY.DEMONS.WHO"
62390 PRINT "{down}WERE.FREED.BY.A.DEMON.WILL.REMAIN,.AND"
62400 PRINT "{down}WILL.BE.A.VERY.LARGE{5°left down #####
        up}.PROBLEM.{down}"
62410 GOSUB 60200

```



FERRY



By John Matarella

Your mission is to ferry supplies through an asteroid belt to a starving, desperate mining base in space. Since the people at the base are dying by the minute, you must try to cross through the belt as fast as you can. You have five remote-controlled ships that travel one at a time across the asteroid belt.

Your ships can travel forward and backward as you try to avoid asteroids. You move forward by

pressing [6], and backward by pressing [4]. When neither key is pressed, the craft doesn't move, but an encounter with an asteroid means the end of that ship. Since the asteroids are always moving, you must be quick to survive.

Your score is based on your speed, how far you get across the asteroid belt, and how many ships survive. Your high score is saved, so you can play several games and watch yourself improve.

```
0 CLR:POKE 59468,12
6 REM FERRY COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="FERRY!":NM$="13":GOSUB 62000
100 CL=59468:MS=2
110 DEF FN R(X)=INT(RND(1)*X+1)
120 R=5:NS=5
130 SD=0:PRINT "{2°down}WANT SOUND?_";:GOSUB 60000
140 IF IN$="" THEN PRINT "{up 12°right}N"
150 IF LEFT$(IN$,1)="Y" THEN SD=16
310 PRINT "{down}HOW HARD_(1=EASY,_9=HARD)_?_";:GOSUB 60000
320 IF IN$="" THEN IN$=STR$(R):PRINT "{up}"TAB(28)R:FOR I=1 TO 500:
NEXT
330 R=VAL(IN$):IF R<1 OR R>9 GOTO 310
335 SF=(R-1)/8*(MS-1)+1
340 TD=0:W=0:YY=0:L=0:T=0:TL=0
350 P=32768+R*WD*2
360 SS=P
370 PRINT "{clr}"
380 FOR I=1 TO 24:PRINT TAB(FN R(37)+1);".{up}":PRINT TAB(
FN R(37)+1);".":NEXT I
400 T=T+1:CC=0
405 POKE 59467,SD:POKE 59464,0:POKE 59466,15
420 CC=CC+1
430 POKE P,62
440 OP=P
450 POKE QP,255:GET A$:A=0:IF A$<>" " THEN A=ASC(A$):IF A$="Q" THEN
POKE 59467,0:GOTO 800
455 XY=15:XX=255:IF A=52 OR A=54 THEN XX=128+SS-P:XY=9:IF A=52 THEN
XY=5
460 POKE 59464,XX:POKE 59466,XY
465 IF A=54 THEN P=P+1
466 IF A=52 THEN P=P-1
480 IF P>=SS+39 THEN 710
```

```

490 IF P<SS THEN P=SS
500 POKE OP,32
520 PRINT TAB(FN R(37)+1);".{up}":PRINT TAB(FN R(37)+1);"."
530 IF PEEK(P)<>46 THEN 420
570 L(T)=P-SS
575 PRINT "{home}";TAB(25);"YOU_BLEW_IT!":PRINT TAB(25);"SHIP.#";T
580 POKE CL,12
590 POKE 59467,SD:POKE 59466,15
600 FOR I=50 TO 250 STEP 3
610 POKE P,42
620 POKE 59464,I
630 POKE P,62
640 NEXT I
670 GOTO 730
710 POKE OP,32:XX=0
715 PRINT "{home}";TAB(25);"SUCCESS!":PRINT TAB(25);"SHIP.#";T
720 POKE QL,24
725 W=W+1:L(T)=40
730 POKE 59466,0:POKE 59467,0
740 POKE P,32
745 GET T$:IF T$<>" " THEN 745
750 D=((P-SS)↑2)/CC:TD=TD+D:GOSUB 1300:PRINT "{home}SCORE:";YY
755 D=TD:GOSUB 1300:PRINT "TOTAL_SO_FAR:";YY
760 PRINT "{home 23°down}PRESS_ANY_KEY_TO.";
765 IF T=NS THEN PRINT "SEE_SUMMARY."
770 IF T<NS THEN PRINT "LAUNCH_NEXT_SHIP."
775 GET T$:IF T$="" THEN 775
777 IF T$="Q" THEN 800
780 IF T<NS THEN P=SS:GOTO 350
800 PRINT "{clr down}YOU_WERE_GIVEN";T;"SHIPS_TO_PILOT_TO"
805 PRINT "THE_ASTEROID_BASE.";W;"SHIP";:IF W<>1 THEN PRINT "S";
808 PRINT "_SURVIVED."
810 D=TD:GOSUB 1300
815 IF W=0 AND SD>0 THEN GOSUB 900
830 PRINT "{2°down}YOUR_SCORE=";YY
835 IF YY>HY THEN HY=YY
837 PRINT "{2°down}HIGHEST_SCORE=_";HY
850 GET T$:IF T$<>" " THEN 850
860 PRINT
870 PRINT "DO_YOU_WANT_TO_PLAY_AGAIN?.";
875 GOSUB 60000
885 IF LEFT$(IN$,1)<>"N"GOTO 310
890 END
900 RESTORE:A=20
905 PRINT "{down}WE_SHALL_NOW_HONOR_THOSE_COMPUTERS_THAT"
906 PRINT "GAVE_THEIR_LIVES_FOR_THE_TRAINING_OF"
907 PRINT "THIS_(AHM...)_PILOT."
910 POKE 59467,16:POKE 59464,0:POKE 59466,104
920 READ C,T,Z
930 T=T*A:Z=Z*A
940 POKE 59464,C:FOR I=1 TO T:NEXT I

```



```

950 POKE 59464,0:FOR I=1 TO Z:NEXT I
960 IF Z>0 THEN 920
970 POKE 59466,0:POKE 59467,0:RETURN
1200 DATA 237,10,1,237,5,1,177,20,5,237,10,1,177,5,1,140,20,5
1210 DATA 237,10,1,177,5,1,140,15,2,237,10,1,177,5,1,140,15,2
1220 DATA 237,10,1,177,5,1,140,25,5,177,10,1,140,5,1,118,25,1
1230 DATA 140,20,1,177,15,1,237,25,1,237,20,1,237,10,1,177,25,0
1300 YY=INT((D*SF)/(39*MS*NS)*1000):RETURN
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$(" _{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT " _{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off} _TO_";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;" _- _CURSOR_#";NM$;
      " _ _ (C) _1981 _ _ _ PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 QP=515:QL=245:CR$=CHR$(13)
60430 IF PEEK(50000) THEN QP=151:QL=216
60450 RETURN
60500 FOR I=1 TO 13:PRINT " {@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;" _ _ BY _JOHN _MATARELLA"
62010 PRINT "{down} _ _ _ _ _CURSOR_#13 _ _ COPYRIGHT _ (C) _1981"
62020 GOSUB 60500
62030 PRINT "{down}CARRY _SUPPLIES _TO _THE _ASTEROID _OUTPOST"
62080 PRINT "{2°down}WANT _DIRECTIONS? _";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=3:GOSUB 60200
62110 PRINT "FAR _OUT _IN _THE _ASTEROIDS, _A _SMALL _GROUP"

```

```

62120 PRINT "{down}OF PEOPLE HAVE SET UP A MINING COMPANY."
62130 PRINT "{down}LIKE MOST OTHER PEOPLE, THEY NEED TO ET{left
ins}A"
62140 PRINT "{down}OCCASIONALLY...YOU HAVE BEEN GIVEN THE"
62150 PRINT "{down}RESPONSIBILITY OF DELIVERING NEEDED"
62160 PRINT "{down}SUPPLIES TO THE BASE OUT IN THE ASTER-"
62170 PRINT "{down}OIDS."
62180 GOSUB 60200
62190 PRINT "SINCE PILOTING THROUGH THE ASTEROID BET{left ins}L"
62200 PRINT "{down}IS HAZARDOUS, AND THERE ARE TOO MANY"
62210 PRINT "{down}PEOPLE INVOLVED TO ALLOW ANY RISK TO"
62220 PRINT "{down}LIFE, REMOTE CONTROL SHIPS{up left ### 2°left
down} ARE USED."
62230 PRINT "{2°down}YOUR TASK IS TO NAVIGATE THE SUPPLY"
62240 PRINT "{down}SHIPS THROUGH THE ASTEROIDS TO THE BAS.{left
ins}E"
62250 PRINT "{down}THERE ARE FIVE SHIPS, AND THEY FLY ONE-"
62260 PRINT "{down}AT-A-TIME (THE EASIER TO CONTROL THEM)."
62270 GOSUB 60200
62280 PRINT "YOUR SCREEN WILL SHOW YOU THE PATH OF A"
62290 PRINT "{down}SHIP, AND THE ASTEROIDS THAT IT PASSES."
62300 PRINT "{down}TO MOVE THE SHIP FORWARD (RIGHT), PRESS"
62310 PRINT "{down}THE_{rvs}6{off left up $
down}_KEY, AND TO MOVE IT BACK (LEFT)"
62320 PRINT "{down}PRESS THE_{rvs}4{off left up $ down}_KEY."
62330 PRINT "{2°down}YOU WILL BE AWARDED A RATING DEPENDING"
62340 PRINT "{down}ON HOW QUICKLY YOU GET THE SUPPLIES"
62350 PRINT "{down}THROUGH, AND HOW FEW SHIPS YOU DESTROY."
62360 PRINT TAB(14); "{2°down}GOOD LUCK!"
62370 GOSUB 60200

```



BAT

By Chris Nadovich



Life isn't easy for a bat, as you'll find out in this game. You have to avoid bouncing off walls, make quick decisions about good-tasting and bad-tasting bugs, keep beating those little wings, and avoid starving to death. As a bat, you need to fly around and eat insects. Some of the insects are tasty, and will help you stay alive. But the so-called "dirt bug" is worthless junk food, and "lint" actually reduces your current calories by half.

The insects have the following values:

☒	Elephant bug	25 calories
♣	Buffet beetle	10 calories
♥	Feast fly	doubles calories
♦	Lint	halves calories
♠	Dirt bug	worthless

To liven things up a bit, the radar of this

species of bat doesn't work very well, so it has evolved a very effective ability to fly into walls in the dark and bounce off.

To control the bat, you use the numeric pad in the usual manner, so that [8] means up, [6] means right, and so on. Note that pressing the button merely applies force; it does not necessarily cause an immediate change in direction. For example, if the bat is moving left, and you hold down [6] to move right, it must first slow down and stop before it starts moving right. That's why this game is so difficult.

To apply a force for a long time, just hold down the button for a while. You don't need to let it up, except to change the direction of the applied force.

Remember, all the time you're flying, you're burning calories. We'll put 50 calories in your tank to start you off.

```

0 CLR:POKE 59468,12
6 REM BAT! COPYRIGHT(C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="BAT!":NM$="14":GOSUB 62000
100 PRINT "{down}WANT SOUND?";:GOSUB 60000
101 IF IN$="" THEN PRINT TAB(12);"{up}N":FOR I=1 TO 500:NEXT
105 PRINT "{clr}"
110 SD=0:IF LEFT$(IN$,1)="Y" THEN SD=16
120 POKE 59467,SD:POKE 59466,15:POKE 59464,0
130 DIM T(29):FOR I=0 TO 29:T(I)=INT(257/(2^(I/12))-1.5):NEXT I
140 Q8=59464
160 D(1)=26:D(2)=18:D(3)=25:D(4)=42:D(6)=41:D(7)=58:D(8)=50:D(9)=57
200 G=9.8:REM GRAVITATIONAL ACCEL.
210 U=.3:REM FRICTION COEFF.
220 A=20:REM THRUST ACCEL.
250 U=1-U:G=G/(9.8*4):A=A/(9.8*4)
500 RESTORE:GF=.142:TP=50
510 TQ=TI:ZQ=59456:ZW=32:S0=8*16*16*16
600 PRINT "{clr &&&&&}CALORIES:.....{&&&&}TIME:.....{&&&&&}";
610 FOR I=1 TO 23:PRINT "{left down &}";:NEXT
620 FOR I=1 TO 39:PRINT "{2°left &}";:NEXT
630 FOR I=1 TO 24:PRINT "{up left &}";:NEXT
640 PRINT "{home}":GOSUB 2000
700 X=20:Y=12:VY=0:VX=0
1000 K=S0+INT(X+.5)+WD*INT(Y+.5)

```

```

1010 R=PEEK(K):TP=TP-GF*(TP/50+1)
1020 IF Q9=1 THEN N=255:Q9=0:GOTO 1040
1030 IF Q9=0 THEN N=127:Q9=1:GOTO 1040
1040 IF R=32 GOTO 1150
1090 IF R=214 THEN TP=TP+25:POKE Q8,T(21):GOTO 1140
1100 IF R=88 THEN TP=TP+10:POKE Q8,T(24):GOTO 1140
1110 IF R=83 THEN TP=TP*2:POKE Q8,T(29):GOTO 1140
1120 IF R=42 THEN POKE Q8,T(1):TP=TP/2
1140 GOSUB 4000
1150 POKE K,N
1180 POKE QP,255:GET Q$:IF Q$="" THEN 1600
1182 IF Q$="Q" THEN 3020
1185 C=ASC(Q$)-48
1190 TP=TP-GF*A
1250 IF C=6 THEN POKE Q8,T(9):VX=VX+1*A:Q=Q-128:GOTO 1600
1300 IF C=2 THEN POKE Q8,T(17):VY=VY+1*A:Q=Q-64:GOTO 1600
1400 IF C=4 THEN POKE Q8,T(12):VX=VX-1*A:Q=Q-32:GOTO 1600
1500 IF C=8 THEN POKE Q8,T(5):VY=VY-1*A
1600 VY=VY+G:NY=Y+VY:NX=X+VX:POKE Q8,0
1700 IF NX>38 THEN NX=38:VY=U*VY:VX=-VX*U
1750 IF NX<1 THEN NX=1:VY=U*VY:VX=-VX*U
1800 IF NY>22 THEN NY=22:VX=U*VX:VY=-VY*U
1850 IF NY<1 THEN NY=1:VX=U*VX:VY=-VY*U
1900 Y=NY:X=NX
1910 IF TP<0 GOTO 3000
1920 N=32:POKE K,N
1930 IF Q9=1 GOTO 1990
1940 POKE Q8,0
1950 PRINT "{home}";TAB(15);INT(TP);"{left}_":PRINT "{home}";
      TAB(28);INT((TI-TQ)/60);"{left}_":
1990 GOTO 1000
2000 FOR I=1 TO 5
2010 READ BG,BN
2015 GB(I)=BG
2020 FOR J=1 TO BN
2030 BX=2+INT(36*RND(1))
2040 BY=2+INT(21*RND(1))
2050 POKE (S0+BX+BY*WD),BG
2060 NEXT
2070 NEXT
2080 DATA 214,5,88,4,42,5,83,4,65,3
2999 RETURN
3000 REM DUST TO DUST
3001 PRINT "{home 10°down}";TAB(14);"{rvs}YOU ARE DEAD{off}"
3002 RESTORE
3003 FOR I=1 TO 5:READ Q5,Q4:NEXT
3005 FOR K=1 TO 11
3006 READ Q5,Q4:Q4=200*Q4
3007 POKE Q8,T(Q5)
3008 FOR J=1 TO Q4:NEXT
3009 POKE Q8,0

```

```

3010 NEXT
3012 DATA 5,4,5,3,5,1,5,4,8,3,7,1,7,3,5,1,5,3,4,1,5,4
3020 GET RJ$:IF RJ$<>" " THEN 3020
3030 PRINT "{clr 2°down}WANT_TO_TRY_AGAIN?";:GOSUB 60000
3040 IF LEFT$(IN$,1)<>"N" THEN 500
3050 POKE 59467,0:PRINT "{clr}":END
4000 BX=2+INT(36*RND(1))
4040 BY=2+INT(21*RND(1))
4050 POKE (S0+BX+BY*WD),GB(INT(RND(1)*5+1))
4999 RETURN
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{to}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_{cursor}#";NM$;
      "._.(C).1981._.PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 QP=515:CR$=CHR$(13)
60430 IF PEEK(50000) THEN QP=151
60440 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_{by}.C.T._NADOVITCH"
62010 PRINT "{down}._.CURSOR.#14._.COPYRIGHT._.(C).1981"
62020 GOSUB 60500
62030 PRINT "{down}REDUCE_THE_LOCAL_INSECT_POPULATION"
62080 PRINT "{2°down}WANT_DIRECTIONS?";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300

```

```

62100 LP=4:GOSUB 60200
62110 PRINT "OVER THE YEARS, RADIO-CONTROLLED MODELS{down}"
62120 PRINT "HAVE BECOME EVER MORE SOPHISTICATED. TE{left
ins}H{down}"
62130 PRINT "LATEST DEVELOPMENTS ARE SO LIFE LIKE. THT{left
ins}A{down}"
62140 PRINT "THEY SEEM ALIVE. YOU WILL SOON HAVE A{down}"
62150 PRINT "CHANCE TO TRY ONE OUT. THE MODEL, CALLD{left
ins}E{down}"
62160 PRINT "THE 'SUPERBAT', COMES IN THE FORM OF A{down}"
62170 PRINT "BAT, AND IS ALSO A GAME OF SKILL. THE{down}"
62180 PRINT "AIR IS FILLED WITH 'INSECTS' FROM A DI-{left
ins}S{down}"
62190 PRINT "PENSER (AVAILABLE AT EXTRA COST), WHICH{down}"
62200 PRINT "THE SUPERBAT MUST EAT TO STAY 'ALIVE'."
62210 GOSUB 60200
62220 PRINT "THE 'INSECTS' HAVE VARIOUS CALORIC{down}"
62230 PRINT "VALUES, AS FOLLOWS:{down}"
62240 PRINT "{rvs V off} - ELEPHANT BUG. 25 CALORIES{down}"
62250 PRINT "{X} - BUFFET BEETLE. 10 CALORIES{down}"
62260 PRINT "{S} - FEAST FLY. DOUBLES CALORIES{down}"
62270 PRINT "{*} - LINT. HALVES CALORIES{down}"
62280 PRINT "{A} - DIRT BUG. WORTHLESS{down}"
62300 PRINT "TO LIVEN THINGS UP, THE SUPERBAT IS{down}"
62310 PRINT "COATED WITH HIGH-RESILIENCY RUBBER, SO{down}"
62320 PRINT "THAT IT BOUNCES EXTREMELY WELL."
62340 GOSUB 60200
62350 PRINT "TO CONTROL THE SUPERBAT, YOU USE THE{down}"
62360 PRINT "NUMERIC PAD IN THE USUAL MANNER, SO THT{left
ins}A{down}"
62370 PRINT
        "{rvs}8{off} MEANS UP, {rvs}6{off} MEANS RIGHT, AND SO ON.{down}"
62380 PRINT "NOTE THAT PRESSING THE BUTTON MERELY{down}"
62390 PRINT "APPLIES FORCE. IT DOES NOT NECESSARILY{down}"
62400 PRINT "CAUSE AN IMMEDIATE CHANGE IN DIRECTION.{down}"
62410 PRINT "FOR EXAMPLE, IF THE SUPERBAT IS MOVING{down}"
62420 PRINT "LEFT, AND YOU HOLD DOWN {rvs}6{off} TO MOVE RIGHT,{left
ins}T{down}"
62430 PRINT "IT MUST FIRST SLOW DOWN AND STOP BEFORE{down}"
62440 PRINT "IT STARTS MOVING RIGHT."
62450 GOSUB 60200
62460 PRINT "TO APPLY A FORCE FOR A LONG TIME, JUST{down}"
62470 PRINT "HOLD THE BUTTON DOWN FOR A WHILE. YOU{down}"
62480 PRINT "DON'T NEED TO LET IT UP EXCEPT TO CHANE{left
ins}G{down}"
62490 PRINT "THE DIRECTION OF THE APPLIED FORCE.{down}"
62500 PRINT
        "ALSO, THE {rvs}1{off}, {rvs}3{off}, {rvs}7{off}, AND {rvs}9
        {off} KEYS DON'T DO{down}"
62510 PRINT "ANYTHING EXCEPT GET IN THE WAY.{down}"
62520 PRINT "{down}REMEMBER THAT GRAVITY WILL EVENTUALLY{down}"

```

```
62530 PRINT "PULL.THE.SUPERBAT.TO.THE.FLOOR.IF.YOU{down}"
62540 PRINT "DON'T.PUSH.IT.UP.OCCASIONALLY."
62550 GOSUB 60200
```



Your car is a white Porsche (shown as a solid white circle), and his is a black Ferrari (shown as a black circle with white border). The track has five lanes which are separated from each other by large steel slabs. In the four shaded areas, the slabs have been removed, allowing you to change lanes. Since your car is always moving, you need only change lanes by using the numeric pad in the usual fashion.

26


```

off ]}"
260 PRINT S$"{ } rvs &&&&&&& off ]}+_+N+A+B+_+{ } rvs &&&&&&&
off ]}"
270 PRINT S$"{ } rvs &&&&&&& off ] rvs}GAME{&&}SCORE{off ] rvs
&&&&&&& off ]}"
280 PRINT S$"{ } rvs &&&&&&& off ]}.....{ } rvs &&&&&&&
off ]}"
290 PRINT S$"{ } rvs &&&&&&& off }_+{-@@@@@@@@@@@= right rvs &&&&&&&
off ]}"
300 PRINT
S$"{ }+{ }+{ }+{ }+{ }+_+_+{&&&&}+_+_+{ }+{ }+{ }+{ }+{ }"
310 PRINT S$"{ }_+{ }_+{ }_+{ }_+{-@@@@@&&&&@@@@@=} _+{ }_+{ }_+{ }_+{ }"
320 PRINT S$"{ }+{ }+{ }+{ }+_+_+_+{&&&&}+_+_+_+{ }+{ }+{ }+{ }"
330 PRINT S$"{ }_+{ }_+{ }_+{-@@@@@@@@@&&&&@@@@@@@@@=} _+{ }_+{ }_+{ }"
340 PRINT S$"{ }+{ }+{ }+_+_+_+_+{&&&&}+_+_+_+_+{ }+{ }+{ }"
350 PRINT S$"{ }_+{ }_+{-@@@@@@@@@&&&&@@@@@@@@@=} _+{ }_+{ }"
360 PRINT S$"{ }+{ }+_+_+_+_+_+{&&&&}+_+_+_+_+_+{ }+{ }"
370 PRINT S$"{ }_+{-@@@@@@@@@@@@@&&&&@@@@@@@@@@@@@=} _+{ }"
380 PRINT S$"{ }+_+_+_+_+_+_+{&&&&}+_+_+_+_+_+_+{ }"
390 PRINT S$"{-@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@= home}"
400 PRINT "{12°down}";TAB(15);MID$(STR$(TR),2)
410 CRT=32768
420 K=CRT+23*WD+15+LEN(S$):P=K+2
425 DF=1
430 D(0)=1:D(1)=-WD:D(2)=-1:D(3)=WD
440 DK=2:DP=0:TK=1:TP=1:V9=43
450 SK=0:SC=0:FP=V3:FK=102
460 POKE P1,0:POKE K,87:POKE P,81:IF P=K THEN 690
470 POKE P1,SD:FOR P9=1 TO 10:NEXT P9:POKE P1,0:SD=0
480 C=PEEK(P+D(DP)):IF C<>43 AND C<>V9 THEN 500
485 IF C=43 THEN SC=SC+10:SD=80:GOTO 495
490 SC=SC+20:SD=40
495 PRINT "{up}"TAB(18)RIGHT$("....."+STR$(SC),8):C=V8
497 IF SC>999 THEN V9=90
500 IF C=93 OR C=64 THEN DP=DP+1:C=V2:IF DP>3 THEN DP=0
510 IF C=87 THEN 690
520 POKE P,FP:FP=C:P=P+D(DP):IF RND(1)>.85 THEN 600
530 C=PEEK(K+D(DK)):IF C=93 OR C=64 THEN DK=DK-1:C=V2:IF DK<0 THEN
DK=3
540 IF C=V8 THEN C=V9
545 IF C=81 THEN 690
550 POKE K,FK:K=K+D(DK):FK=C
552 IF C<>V3 AND C<>V4 THEN TL=DF:GOTO 600
556 TL=TL-1:IF TL<0 THEN 600
560 IF C=V3 AND TK<TP THEN K=K+V7*D(DK):TK=TK+V1
570 IF C=V3 AND TK>TP THEN K=K-V7*D(DK):TK=TK-V1
580 IF C=V4 AND TK<TP THEN K=K-2*D(DK)/WD:TK=TK+V1
590 IF C=V4 AND TK>TP THEN K=K+2*D(DK)/WD:TK=TK-V1
600 IF FP<>V3 AND FP<>V4 THEN 680
601 POKE QP,255:GET K$:IF K$="" THEN K$="5"
603 IF K$="Q" OR K$="_" THEN 780

```

```

605 KB=ASC(K$)-48:MV=0
610 IF KB<2 OR KB>8 OR (KB AND 1) THEN 680
620 IF KB=8 AND FP=V3 THEN JS=P-WD*2:TT=TP+SGN(D(DP)):MV=1
630 IF KB=2 AND FP=V3 THEN JS=P+WD*2:TT=TP-SGN(D(DP)):MV=1
640 IF KB=4 AND FP=V4 THEN JS=P-2:TT=TP-SGN(D(DP)):MV=1
650 IF KB=6 AND FP=V4 THEN JS=P+2:TT=TP+SGN(D(DP)):MV=1
655 IF MV=0 THEN 680
660 IF MV THEN SD=151
663 IF ABS(P-JS)<>2 AND ABS(P-JS)<>WD*2 THEN STOP
670 IF PEEK(JS)=FP THEN TP=TT:P=JS
680 DF=1+INT(SC/500)
685 GOTO 460
690 X=P-CRT:PRINT "{home}";
700 Y=INT(X/WD):X=X-Y*WD
710 FOR A=1 TO Y:PRINT "{down}";:NEXT
720 PRINT TAB(X-2)"{right CV}-{up 3°left M}N 2°down 3°left N}M
    home 3°down}"
730 POKE P2,15
740 FOR A=1 TO 25:POKE P1,255:POKE P1,250:POKE P1,253:NEXT
750 POKE P1,0:POKE P3,16:POKE P2,15
760 TM=TI+150
770 IF TI<TM THEN 770
780 GET A$:IF A$<>" " THEN 780
790 IF SC>HS THEN HS=SC
800 CS=CS+SC
810 PRINT "{clr 5°down}_YOUR_SCORE_FOR_GAME"TR"WAS"SC;"{left},"
820 PRINT
    "{down}_FOR_AN_AVERAGE_OF"INT(CS/TR+.5)"POINTS/GAME.{down}"
830 PRINT "_YOUR_HIGH_SCORE_SO_FAR_IS"HS;"{left}."
840 PRINT "{3°down}_DO_YOU_WISH_TO_PLAY_AGAIN?_";:GOSUB 60000
850 A$=LEFT$(IN$,1)
860 IF A$="N" THEN PRINT "{clr}":GOTO 880
870 FOR A=1 TO 200:NEXT:TR=TR+1:GOTO 140
880 POKE P2,0:POKE P3,0:POKE 59468,12
890 END
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000

```

```

60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_{-}_CURSOR_{#}";NM$;
    "_{(C)}_{1981}_{PAGE}";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}";GOSUB 60400:GOTO 100
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 QP=515:QV=1:CR$=CHR$(13)
60430 IF PEEK(50000) THEN QP=151:QV=2
60440 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;
    "_{BY}_{MALCOLM}_{MICHAEL}"
62010 PRINT "{down}_{CURSOR}_{#16}_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62030 PRINT "{down}_{A}_{RACE}_{AGAINST}_{DESTRUCTION}..."
62080 PRINT "{2°down}PRESS_{rvs}RETURN{off}_{TO}_{BEGIN}"
62090 GET T$:IF T$="" THEN 62090
62100 GOTO 60300

```



PUZZLES



HANOI

By Glen Fisher

Somewhere, secluded from modern civilization, a group of priests has been moving golden disks from one great golden pedestal to another. When they're done, so the story goes, the world will end.

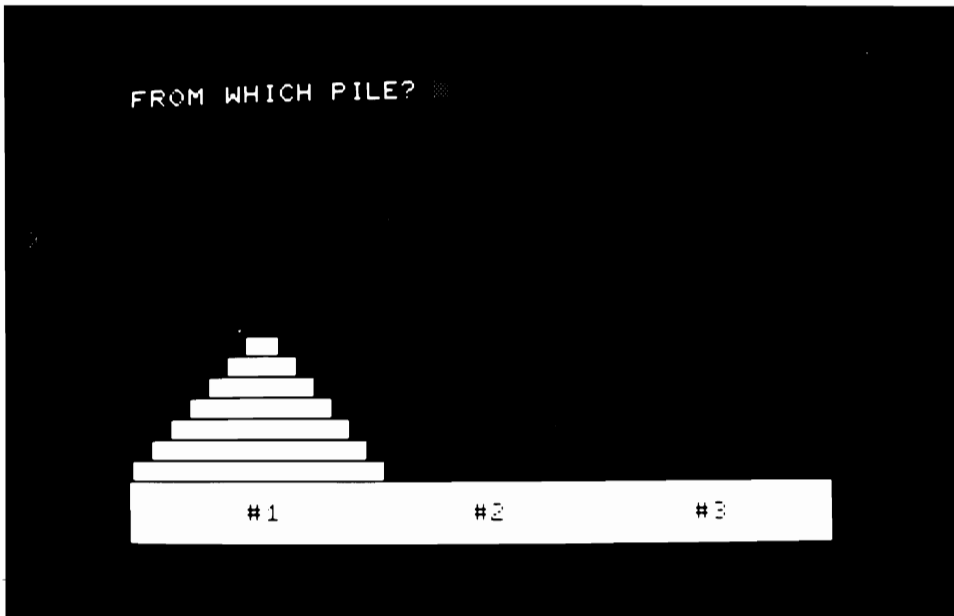
In the legend, there were three pedestals and, some say, 64 disks in a regular series (each disk slightly larger than the last). Initially, all the disks were on one of the three pedestals (with the largest at the bottom and the smallest at the top). The priests moved the whole pile to another pedestal, one disk at a time. There was one restriction placed on them: a disk could be placed only on top of a larger disk, never on a smaller one. With that restriction, assuming they could move one disk per *second*, it would take them more

than a half trillion *years* to move the pile of 64 disks.

HANOI is a much-simplified version of that task, with a much less cosmic ending; when you finish, only the program ends. However, the restrictions are the same. You may choose from two to seven disks in the pile. To make a move, just indicate from which pile you want to take the top disk, and on which pile you want the disk placed. HANOI won't allow you to make an illegal move.

If you want to give up, type "Q" (for quit). When you're done, you'll be told how many moves you took, as well as the minimum number of moves necessary.

Oh yes, the program is called HANOI because the priests were once thought to be in Hanoi.

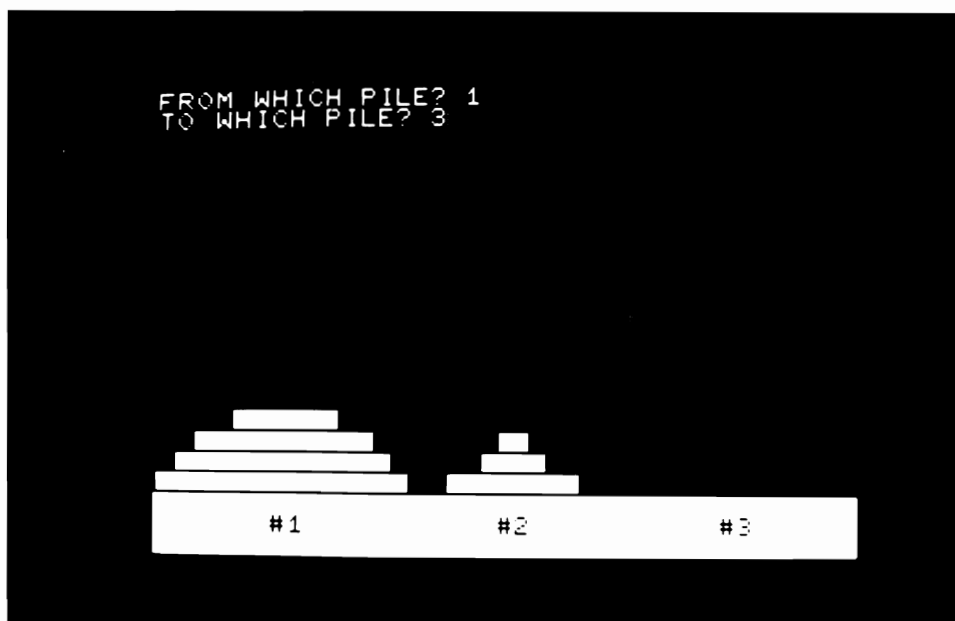


```
0 CLR:POKE 59468,12:NM$="5"
6 REM HANOI COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 GOSUB 62000
100 T$="{off}_ {off $ off / off 9 off " rvs 8 rvs 7 rvs #}"
105 B$="{rvs $ rvs / rvs 9 rvs " off 8 off 7 off # off}_"
110 DIM TP$(8),T$(8),BT$(8),B$(8)
115 CR$=""
120 FOR I=1 TO 8
130 C$=MID$(T$,2*I,1):C$=C$+C$+C$:C$=C$+C$+C$:C$=LEFT$(C$+C$,14)
```

```

140 TP$(I)=MID$(T$,2*I-1,1)+C$
150 C$=MID$(B$,2*I,1):C$=C$+C$+C$:C$=C$+C$+C$:C$=LEFT$(C$+C$,14)
160 BT$(I)=MID$(B$,2*I-1,1)+C$
170 NEXT I
200 L$="{rvs}_ {rvs 4 rvs ! off * off}_ "
210 R$="{off}_ {off 4 off ! rvs * rvs}_ "
220 DIM LF$(5),RT$(5)
230 FOR I=1 TO 5:LF$(I)=MID$(L$,2*I-1,2):RT$(I)=MID$(R$,2*I-1,2):
NEXT I
500 PRINT :PRINT "HOW MANY DISKS (UP TO 7) ? ";:GOSUB 60000:PRINT
510 IF IN$="" THEN PRINT "LET'S USE 3":IN$="3":FOR I=1 TO 500:NEXT
520 FOR I=1 TO LEN(IN$)
530 :T$=MID$(IN$,I,1):IF T$>="0" AND T$<="9" THEN 550
540 PRINT "NO ";T$;"'S PLEASE.":GOTO 500
550 NEXT I
560 N=VAL(IN$):IF N>7 THEN PRINT "I CAN'T HANDLE MORE THAN 7.":
GOTO 500
570 IF N<2 THEN PRINT "DON'T BE RIDICULOUS!":GOTO 500
900 DN$="{home 20°down}"
910 RT$="{14°right}"
1000 FOR I=1 TO 3:FOR J=0 TO 7:P(I,J)=0:NEXT J:NEXT I
1100 PRINT "{clr}";DN$;:P(1,0)=N

```



```

1110 FOR I=1 TO 3:
PRINT "{rvs}.....{left
ins}_":NEXT I
1120 PRINT DN$;"{down rvs 6°right}#1{11°right}#2{11°right}#3"
1150 PRINT DN$;LEFT$("{7°up}",N);
1160 FOR I=1 TO N
1170 PRINT LEFT$("{off}" + RT$,7-I+1);LEFT$("{rvs L}" + BT$(1),2*I+1);
"{:}"
1180 P(1,N-I+1)=I*2
1190 NEXT I

```

```

1800 MV=0
1900 TM=TI+60
1910 IF TI<TM THEN 1910
1950 GOSUB 8000
2000 PRINT :PR$="FROM WHICH PILE?":GOSUB 5000:IF IN$="" THEN 1950
2010 F=VAL(IN$)
2020 IF P(F,0)<1 THEN PRINT "THAT PILE IS EMPTY.":GOTO 1900
2050 PR$="TO WHICH PILE?":GOSUB 5000:IF IN$="" THEN 1950
2060 T=VAL(IN$)
2070 IF F=T THEN PRINT "THAT MOVE DOES NOTHING.":GOTO 1900
2080 IF P(T,0)=0 THEN 2100
2090 IF P(F,P(F,0))>P(T,P(T,0)) THEN PRINT "IT WON'T FIT.":GOTO 1900
2100 FC=P(F,0):FW=P(F,FC):TC=P(T,0):TW=P(T,TC):MV=MV+1
2110 X=1+13*(F-1)+7-FW/2
2120 PRINT DN$;LEFT$("{9°up}",FC+1);LEFT$("{off}"+RT$+RT$+RT$,X);
2125 LF$=LEFT$("{down 14°left}",FW+1)
2130 FOR I=1 TO 8:T$(I)=LEFT$(TP$(I),1+FW):B$(I)=LEFT$(BT$(I),1+FW):
NEXT I
2135 HT=0:J=1:IF F>T THEN J=-1
2140 FOR I=F TO T STEP J
2145 IF P(I,0)>HT THEN HT=P(I,0)
2150 NEXT I
2155 IF HT=P(T,0) THEN HT=HT+1:GOTO 2165
2160 IF ABS(F-T)>1 THEN IF HT=P(2,0) THEN HT=HT+1
2165 FOR I=FC TO HT
2170 :FOR J=1 TO 8
2175 :PRINT T$(J);LF$;B$(J);LF$;"{2°up}";
2180 :NEXT J
2185 :PRINT "{up}";
2190 NEXT I
2195 PRINT "{down}";
2200 L$=LEFT$("{16°left}",1+FW):R$=LEFT$(RT$,FW-1)
2201 PRINT "{rvs}";LEFT$(".....",FW);"{off}";
LEFT$("{16°left}",FW);
2205 IF F>T THEN 2250
2210 FOR I=F*13 TO T*13-1
2215 :FOR J=1 TO 5
2220 PRINT LF$(J);R$;RT$(J);L$;
2225 NEXT J
2226 PRINT "{right}";
2230 NEXT I
2235 GOTO 2300
2250 FOR I=T*13 TO F*13-1
2255 PRINT "{left}";
2260 :FOR J=5 TO 1 STEP -1
2265 ::PRINT LF$(J);R$;RT$(J);L$;
2270 :NEXT J
2275 NEXT I
2300 FOR I=HT-1 TO TC STEP -1
2310 :FOR J=8 TO 1 STEP -1
2320 :PRINT T$(J);LF$;B$(J);LF$;"{2°up}";

```



```

2330 :NEXT J
2340 :PRINT "{down}";
2350 NEXT I
2410 PRINT LEFT$("{rvs L}" + BT$(1), FW+1); "{:}"
2500 P(T,0)=P(T,0)+1
2510 P(T,P(T,0))=P(F,P(F,0))
2520 P(F,0)=P(F,0)-1
2600 IF P(2,0)<>N AND P(3,0)<>N THEN 1900
2700 GOSUB 8000
2710 PRINT "{3°down}.....{,"""""""""""";}"
2720 PRINT ".....{rvs !}YOU DID IT!!{off !}"
2730 PRINT ".....{< rvs """""""""" off >}"
2740 PRINT "{down}IT TOOK YOU";MV;LEFT$("MOVES",5+(MV=1));
      "_TO_DO_IT."
2750 T=2↑N-1:PRINT "{down}THE SHORTEST POSSIBLE SOLUTION"
2760 PRINT "NEEDS";T;LEFT$("MOVES",5+(T=1));"."
2770 PRINT "{down}PLAY AGAIN?_";GOSUB 60000:PRINT "{clr}"
2780 IF LEFT$(IN$,1)="N" THEN END
2790 CLR:GOTO 100
2800 END
5000 PRINT PR$;
5010 GOSUB 60000:PRINT :IF IN$="" THEN RETURN
5015 IF LEFT$(IN$,1)="Q" THEN END
5016 IF LEN(IN$)>1 THEN 5030
5020 IF IN$>="1" AND IN$<="3" THEN RETURN
5030 PRINT "THERE'S NO SUCH PILE."
5040 FOR I=1 TO 500:NEXT I
5050 PRINT "{up}....."
5060 PRINT "{2°up}";PR$;
      LEFT$(".....",LEN(IN$))
5070 PRINT "{up}";GOTO 5000
8000 PRINT "{home}";
8010 FOR I=1 TO 4:PRINT ".....":
      NEXT I
8020 PRINT "{home}";
8030 RETURN
9000 PRINT "{home}"
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>15 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010

```

```

60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN GOTO 60250
60205 GOSUB 60500
60210 PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;".-CURSOR.#";NM$;
      ".(C).1981....PAGE";PN;"{down}"
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:CR$=CHR$(13):GOTO 100
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PG$="HANOI":LP=1:CR$=CHR$(13)
62020 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;".BY.GLEN.FISHER"
62021 PRINT "{down}....CURSOR.#5..COPYRIGHT.(C).1981"
62022 GOSUB 60500
62025 PRINT "{down}ANIMATED.TOWERS.OF.HANOI.PUZZLE"
62030 PRINT "{2°down}WANT.DIRECTIONS?";:GOSUB 60000
62040 IF LEFT$(IN$,1)="Y" THEN GOTO 62060
62050 PRINT "{clr 3°down}":RETURN
62060 GOSUB 60200
62070 PRINT "THE.OBJECT.OF.THIS.PUZZLE.IS.TO.MOVE"
62080 PRINT "A.PILE.OF.DISKS..HOWEVER,.A.LARGER.DISK"
62090 PRINT "MAY.NEVER.BE.PLACED.ON.TOP.OF.A.SMALLER"
62100 LP=1:PRINT "DISK."
62110 PRINT "{down}THE.PET.WILL.ASK.YOU.HOW.MANY.DISKS"
62120 PRINT "YOU.WISH.TO.WORK.WITH..(BEGINNERS:"
62130 PRINT "TRY{.}3.OR.4...)..IT.WILL.THEN.DRAW"
62140 PRINT "A.PICTURE.OF.THE.DISKS,.AND.ASK:"
62150 PRINT "'FROM.WHICH.PILE?'..ENTER.THE.NUMBER"
62160 PRINT "OF.THE.PILE.FROM.WHICH.YOU.WANT.TO"
62170 PRINT "TAKE.THE.TOP.DISK..THEN,.YOU.ENTER"
62180 PRINT "THE.NUMBER.OF.THE.PILE.WHERE.YOU"
62190 PRINT "WANT.TO.PLACE.THAT.DISK."
62200 PRINT "{down}REMEMBER.-.YOU.ALWAYS.MOVE.ONE.DISK"
62210 PRINT "AT.A.TIME!.(THE.PET.WON'T.LET.YOU.MAKE"
62220 PRINT "ILLEGAL.MOVES)."
62230 GOSUB 60200

```

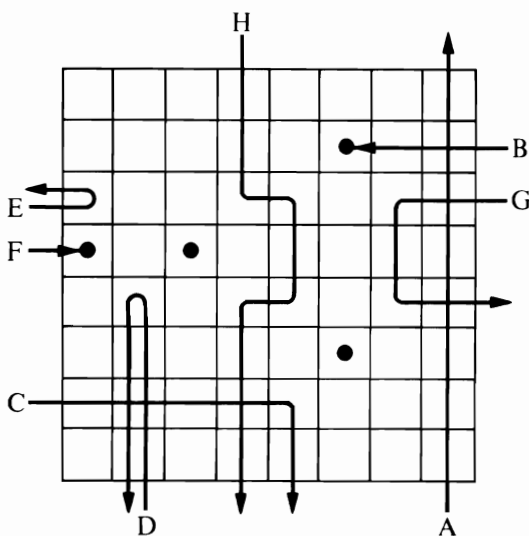


BOX

By Howard Arrington

The semiconductor industry is always looking for new materials with which to make its ever-smaller products. As you may know, integrated circuit chips are made by taking very pure silicon and adding very specific impurities in very particular patterns. In BOX, you have a small piece of a silicon chip, and must determine the exact position, within an 8×8 area, of the atoms from the added impurity. Naturally, since atoms are far too small to see, you must use some other tool to find them.

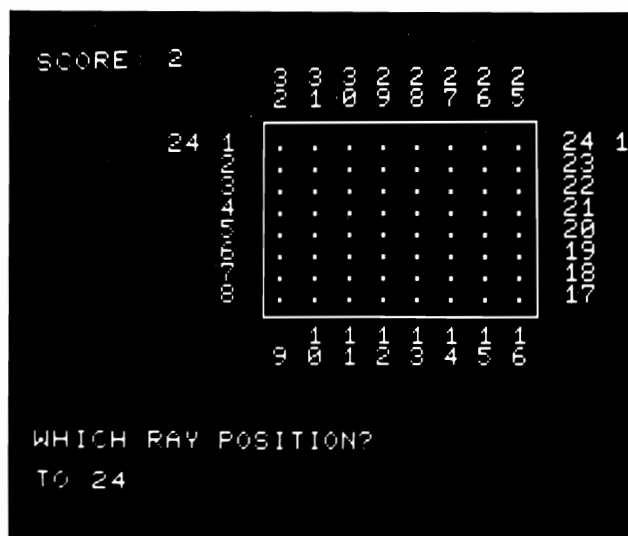
Your tools in this game are a particle “gun” (which shoots a high-energy ray into the chip) and a set of detectors to tell where the ray comes out again, if it does. Since the silicon is so pure to start with, it forms very perfect crystals. As a result, the number of ways a ray can interact with an impurity atom are few, and are shown in the diagram below. The drawing shows a sample 8×8 board, with the atoms marked, and several rays shot into it.



- A) If a shot hits nothing, it goes straight through and comes out the other side.
- B) If a shot hits an atom straight on, it is absorbed by the nucleus, and vanishes. This is a *hit*.
- C) A shot that passes next to an atom is *deflected* 90 degrees away from the atom. Note that it never gets to be even with the atom.

- D) A shot aimed between two atoms one space apart is *reflected* back to its starting point. (It gets deflected 90 degrees twice, resulting in a 180-degree turnaround.)
- E) A shot just to one side of an atom (at the edge) is also *reflected*.
- F) A shot directly at an atom (at the edge) is a *hit*, just as any other head-on shot.
- G) Shots may interact with more than one atom. Here is a shot that gets deflected twice, and comes out the same side it came in.
- H) This is another multiple interaction. However, the results of this one appear the same as no interaction at all. The shot *appears* to go straight through, even though it is deflected by three atoms (twice by one of them).

When the game starts, you'll be asked how many atoms (from 1 to 16) should be put into the



8×8 area. You must then find them by shooting rays.

To make a shot, type in the number of the spot where you want the ray to enter. The result will be shown next to the entry spot by the following: an “H” for a hit, an “R” for a reflection, and the number of the exit spot for anything else. (In the last case, the exit spot will be marked with the

number of the entry spot, as well.)

When you've figured out where an atom is, you should mark it. To do so, press [RETURN] when asked for an entry spot. The PET will ask you for the row and column of the atom (as with shots, use the numbers along the edge). If you're right, the atom will be marked. If you're wrong, the PET will say so. The game is over when

you've marked all the atoms.

To quit and to find out where the atoms are, enter [0] (zero) for the entry point of a shot. A score is being kept during all of this. You are given one point for each hit or reflection, two points for any other result, and five points for each wrong guess of an atom's location. The lower the score, the better you are.

```

0 CLR:POKE 59468,12:PG$="BOX":GOSUB 62000
6 REM COPYRIGHT(C)1981 THE CODE WORKS
100 GOSUB 60400:REM 10/6/81
102 DIM L(32):FOR I=0 TO 7:L(I+1)=I*WD:NEXT I:FOR I=0 TO 7:
    L(I+9)=I*2+WD*12+5:NEXT I
105 FOR I=0 TO 7:L(I+17)=WD*7+25-I*WD:NEXT I:FOR I=0 TO 7:
    L(I+25)=- (WD*5)+19-I*2:NEXT I
110 DATA 487,489,491,493,495,497,499
115 DATA 305,265,225,185,145,105,65,25,-181,-183,-185,-187,-189,
    -191,-193,-195
125 DEF FN R(Z)=INT(8*RND(1)+1)
135 CR$=CHR$(13):A=CRT+WD*6+7
137 W$="....."
140 PRINT :PR$="{up}HOW MANY ATOMS?":GOSUB 4000:IF IN$<>" THEN
    N=IN:GOTO 150
145 PRINT "I'LL ASSUME FIVE":N=5
150 K=0:HA=N:S=0
151 H$="{home 20°down}"
155 RS=22*WD+CRT
160 IF N<1 THEN 3500
161 IF N>16 THEN PRINT "16 IS MAXIMUM{up 2°right}"W$"{up}":GOTO 140
165 FOR J=0 TO 9:FOR I=0 TO 9:B(I,J)=0:NEXT I:NEXT J
175 FOR I=1 TO N
180 X=FN R(1):Y=FN R(1)
185 IF B(X,Y)<>0 THEN 180
190 B(X,Y)=1:NEXT I
195 GOSUB 1000
200 GOSUB 2000:IF K<>Z THEN R=K:K=Z:Z=R:GOSUB 2000
201 PRINT H$"{15°right}"W$
205 PR$="{up}WHICH RAY POSITION?":GOSUB 4000:IF IN$="" THEN IN=-1
206 PRINT W$:R=IN:K=R:IF R<1 THEN 3000
207 PRINT W$,"{up}"
208 IF R>32 THEN 201
210 ON (R-1)/8+1 GOTO 250,260,270,280
220 PRINT "ERROR.....":GOTO 200
250 X=0:Y=R:U=1:V=0:GOTO 300
260 X=R-8:Y=9:U=0:V=-1:GOTO 300
270 X=9:Y=25-R:U=-1:V=0:GOTO 300
280 X=33-R:Y=0:U=0:V=1
300 X1=X+U:Y1=Y+V
310 IF U=0 THEN X2=X1-1:X3=X1+1:Y2=Y1:Y3=Y1:GOTO 330
320 Y2=Y1-1:Y3=Y1+1:X2=X1:X3=X1
330 ON 8*SGN(B(X1,Y1))+SGN(B(X2,Y2))+2*SGN(B(X3,Y3))+1 GOTO 400,
    410,420,410

```

```

340 S=S+1:PRINT "HIT.....":Z=K:GOTO 200
400 X=X1:Y=Y1:GOTO 500
410 Z=1:GOTO 450
420 Z=-1
450 IF U=0 THEN U=Z:V=0:GOTO 500
460 U=0:V=Z
500 ON (X+15)/8 GOTO 550,520,560
510 STOP
520 ON (Y+15)/8 GOTO 570,300,580
530 STOP
550 Z=Y:GOTO 600
560 Z=25-Y:GOTO 600
570 Z=33-X:GOTO 600
580 Z=8+X
600 IF Z=R THEN PRINT "REFLECTION":S=S+1:GOTO 200
610 S=S+2:PRINT "TO";Z;"{left}.....":GOTO 200
700 A=A-WD+4:D1=87
710 FOR J=1 TO 8:FOR I=1 TO 8
720 IF B(I,J)<>1 THEN 740
730 T=A+I*2+J*WD:POKE T,D1
740 NEXT I:NEXT J
750 PRINT H$;W$
760 PRINT W$:PRINT W$;"{3°up}":GOTO 135
1000 PRINT "{clr 3°down}.....3.3.3.2.2.2.2.2"
1020 PRINT TAB(13);"2.1.0.9.8.7.6.5"
1030 PRINT TAB(12);"{000000000000000000.}"
1040 FOR I=1 TO 8
1050 PRINT ".....";RIGHT$(STR$(I),2);"_{1}}.....{1}}_";
1055 PRINT RIGHT$(STR$(25-I),2)
1060 NEXT I
1070 PRINT TAB(12);"{-000000000000000000=}"
1080 PRINT TAB(13);"_.1.1.1.1.1.1.1"
1090 PRINT TAB(13);"9.0.1.2.3.4.5.6"
1100 RETURN
2000 PRINT "{home 2°down}SCORE: ";S;"{left}._."
2005 IF K=0 THEN RETURN
2010 X=PEEK(RS):ZZ=L(K)+A
2020 IF X=20 THEN X$=RIGHT$(STR$(Z),2):GOTO 2100
2030 POKE ZZ+1,X:RETURN
2100 D1=ASC(MID$(X$,1,1)):D2=ASC(MID$(X$,2,1))
2105 IF K>8 THEN 2140
2110 POKE ZZ,D1:POKE ZZ+1,D2:RETURN
2140 IF K>16 THEN 2200
2148 IF D1=32 THEN D1=D2:D2=32
2150 POKE ZZ+1,D1:POKE ZZ+WD+1,D2:RETURN
2200 IF K>24 THEN 2300
2208 IF D1=32 THEN D1=D2:D2=32
2210 POKE ZZ+1,D1:POKE ZZ+2,D2:RETURN
2300 POKE ZZ-WD+1,D1:POKE ZZ+1,D2:RETURN
3000 IF R=0 THEN 700
3005 PRINT H$;W$

```

```

3006 PRINT W$:PRINT W$:PRINT W$;"{3°up}"
3010 PR$="{up}WHICH_ROW?_":GOSUB 4000:IF IN$="" THEN PRINT :
    GOTO 3060
3012 I=IN:IF I>16 AND I<25 THEN I=25-I
3015 IF I<1 OR I>8 THEN PRINT "{up 7°right}"W$:GOTO 3010
3016 PRINT
3020 PR$="{up}WHICH_COLUMN?_":GOSUB 4000:IF IN$="" THEN 3060
3021 J=IN:IF J>8 AND J<17 THEN J=41-J
3025 IF J<25 OR J>32 THEN PRINT "{up 9°right}"W$:GOTO 3020.
3030 J=33-J
3035 IF J<1 OR J>8 THEN 3010
3040 IF B(J,I)=1 THEN POKE A-WD+4+J*2+I*WD,81:B(J,I)=2:HA=HA-1:
    GOTO 3060
3045 IF B(J,I)<>0 THEN PRINT "THAT_ONE_AGAIN???{up}":GOTO 3055
3050 PRINT "WRONG!{up}":S=S+5
3055 FOR I=0 TO 1500:NEXT I
3060 PRINT "{2°up}";W$:PRINT W$:PRINT W$
3065 IF HA=0 THEN 700
3070 K=0:Z=0:GOTO 200
3500 PRINT "{clr}":END
4000 PRINT PR$;:GOSUB 60000:IF IN$="" THEN RETURN
4010 FOR Z=1 TO LEN(IN$):T$=MID$(IN$,Z,1)
4020 IF T$=>"0" AND T$=<"9" THEN NEXT Z:IN=VAL(IN$):RETURN
4030 PRINT "NO_";T$;"'S_PLEASE.":PRINT "{2°up}"W$ LEFT$(W$,15):
    GOTO 4000
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20):PRINT "...{3°left}";
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_&","ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "...{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>18 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "{down}PRESS_{rvs}RETURN{off}_TO_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_CURSOR_#6";NM$;
    "...(C)_1981....PAGE";PN;"{down}"

```

```

60265 GOSUB 60500
60270 RETURN
60400 CRT=32768:PRINT "{clr}"
60410 WD=40:POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:
PRINT CHR$(142)
60415 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;
"._BY._HOWARD._ARRINGTON"
62010 PRINT "{down}.....CURSOR.#6...COPYRIGHT.(C).1981"
62013 GOSUB 60500
62015 PRINT "{down}FIND.THE.HIDDEN.ATOMS{down}"
62020 PRINT "{down}WANT.INSTRUCTIONS?.";:GOSUB 60000:IF LEFT$(IN$,
1)="Y" THEN 62040
62030 PRINT "{clr}":RETURN
62040 LP=4:GOSUB 60200
62100 PRINT "YOU.WILL.SHORTLY.SEE.AN.8X8.BOX.IN"
62110 PRINT "WHICH.SEVERAL.ATOMS.ARE.HIDDEN..YOUR"
62120 PRINT "TASK.IS.TO.FIND.THE.ATOMS..YOU.HAVE.AT"
62130 PRINT "YOUR.DISPOSAL.A.RAY.PROJECTER,WHICH"
62140 PRINT "YOU.CAN.USE.TO.SEND.RAYS.INTO.THE.BOX"
62150 PRINT "FROM.ANY.OF.32.SPOTS.ALONG.THE.SIDES"
62160 PRINT "OF.THE.BOX."
62165 PRINT "{down}ANY.GIVEN.RAY.CAN.MEET.ONE.OF.THREE"
62170 PRINT "FATES:..IT.MAY.HIT.AN.ATOM,AND.BE"
62180 PRINT "ABSORBED;..IT.MAY.BE.REFLECTED,AND.COME"
62190 PRINT "OUT.WHERE.IT.WENT.IN;OR.IT.MAY.BE"
62200 PRINT "DIVERTED.AND.COME.OUT.SOMEWHERE.ELSE."
62210 PRINT "YOU.WILL.BE.TOLD.WHAT.HAPPENS,AND"
62220 PRINT "WHERE.THE.RAY.COMES.OUT.(IF.IT.DOES). "
62230 PRINT "{down}ON.THE.NEXT.PAGE, YOU'LL.SEE.THE.RULES"
62235 PRINT "CONTROLLING.THE.PATH.OF.A.RAY."
62240 GOSUB 60200
62250 PRINT "{down}1:..A.{ }RAY.THAT.HITS.AN.ATOM.DEAD.ON.IS"
62260 PRINT "...ABSORBED,AND.NEVER.COMES.OUT...{}{}"
62265 PRINT TAB(34);"{@=}._{-@}"
62270 PRINT "2:..A.RAY.THAT.HITS.AN.ATOM.TO.ONE...O"
62280 PRINT "...SIDE.IS.DIVERTED.90.DEGRESS.TO.{@}._{0@}"
62290 PRINT "...THAT.SIDE.";TAB(35);"{}{}"
62300 PRINT "{down}3:..A.RAY.THAT.PASSES.BETWEEN.TWO...O..O"
62310 PRINT "...ATOMS.ONE.SQUARE.APART.IS.RE-...{0.}"
62315 PRINT "...FLECTED,AND.RETRACES.ITS.PATH...{}{}"
62320 PRINT "{down}4:..A.RAY.THAT.ENTERS.ALONGSIDE...O{0.}"
62330 PRINT "...AN.ATOM.IS.REFLECTED.....{}{}"
62340 PRINT "{down}5:..AT.ALL.OTHER.TIMES.A.RAY.FOLLOWS.A"
62350 PRINT "...STRAIGHT.PATH."
62360 GOSUB 60200
62370 PRINT "THE.PLACES.YOU.CAN.SEND.IN.A.RAY.ARE"
62380 PRINT "NUMBERED.FROM{right}1.TO{ }32..TO.SEND.A.RAY"
62390 PRINT "IN,TYPE.THE.NUMBER.OF.THE.PLACE.YOU"
62400 PRINT "WANT.IT.TO.ENTER.AT..IT.WILL.START.OUT"

```

```
62410 PRINT "TRAVELLING_VERTICALLY_OR_SIDEWAYS,"
62420 PRINT "DEPENDING_ON_WHICH_SIDE_IT_STARTS_AT."
62430 PRINT "{down}YOU_WILL_BE_TOLD_WHETHER_THE_RAY_HITS"
62440 PRINT "AN_ATOM,{ }IS_REFLECTED,_OR_COMES_OUT."
62450 PRINT "SHOULD_IT_COME_OUT,_YOU'LL_BE_TOLD"
62460 PRINT "WHERE...TO_GIVE_UP_(AND_FIND_OUT_WHERE"
62470 PRINT "THOSE_LOUSY_ATOMS_WERE),_TYPE_A_0"
62480 PRINT "(ZERO)_AS_THE_RAY_ENTRANCE._TO_GUESS"
62490 PRINT "AN_ATOM'S_LOCATION,_JUST_HIT_{rvs}RETURN{off}."
62500 GOSUB 60200
62510 PRINT "YOU_ARE_GIVEN_A_SCORE_ACCORDING_TO_HOW"
62520 PRINT "WELL_YOU_DO._FOR_EACH_RAY_THAT_GOES_IN"
62530 PRINT "YOU_GET_ONE_POINT._IF_IT_COMES_OUT"
62540 PRINT "AGAIN,{ }YOU_GET_ANOTHER_POINT._FOR_EACH"
62550 PRINT "WRONG_GUESS_YOU_MAKE,_YOU_GET_FIVE"
62560 PRINT "POINTS...A_LOW_SCORE_INDICATES_HIGH"
62570 PRINT "SKILL."
62630 PRINT "{2°down}.....HAVE_FUN!!{2°down}"
62640 GOSUB 60200
```




MIND

By Glen Fisher

Mind is our version of the popular game “Master Mind”¹, which is a game played with colored pegs on a board. One player invents a color pattern, and the other player tries to guess the pattern. In the computer version, since we can’t use actual pegs, we just print the color names on the screen. The PET will choose the color pattern, and it is your job to discover the pattern.

You enter your guess using the [R], [O], [Y], [G], [B], and [V] (for Red, Orange, Yellow, Green, Blue, and Violet) keys. As you press them, the appropriate color names will be printed on the screen. At its simplest, entering your guess involves just pressing four keys, and then [RETURN]. If you press the wrong key, just press [DEL] (or [LEFT]) to back up the cursor, and press the key you intended. [SPACE] or [RIGHT] move the cursor to the right. Finally, the [?] key means “I give up. What’s the answer?”

Once you’ve pressed [RETURN], the PET will tell you how many “hits” or “misses” you got. You get a *hit* when a color in your guess matches (in color *and* position) a color in the pattern. For example:

pattern: R G B V

guess: B G R O

For that guess, you would get one hit, for the green. Neither red nor blue would be a hit, because they’re in the wrong places. You get a *miss* when one of the colors in your guess matches one of the colors in the pattern, but is in the wrong place. In the above example, red and blue would be misses, since the pattern has both red and blue in it, but in your guess they’re not where they are in the pattern.

There are two complications. First, the PET only tells you *how many* hits and misses you got. It doesn’t say which colors in your guess gave you the hits or misses. In the above example, you would merely be told “one hit, two misses,” with no clue as to which colors were involved. It is

nonetheless possible to figure out the pattern. You make several guesses, and figure out what pattern of colors could produce the hits and misses you’ve been getting. For example, with the pattern above, if you change your guess to

B O R G

you will get “0 hits, 3 misses.” You now know that either Green or Orange was the hit, because they’re the only colors you changed, and the hit went away.

The other complication is that the same color can be used several times in the pattern. Thus, the pattern could be R O G R, with two Reds in it. The thing to be aware of is that any given peg in the guess can make a hit or a miss, but never both. Likewise with the pattern, one peg can only be one hit or miss. For example, with the pattern R O G R, a guess of R Y B O would give you one hit (for red) and one miss (for orange). The other red wouldn’t count.

For a guess of R Y R B, you’d also get one hit (for red) and one miss (for the *other* red). A guess of Y R G B would give a hit (for green) and a miss (for red). You only get one miss because you only guessed one red. A guess of G R B G would give no hits and two misses, one for red, and one for green. There’s only one green in the pattern, so only one of your greens is a miss. The other green is ignored.

To show you how many hits and misses you got, the PET prints dots beside your guess. A white dot marks a hit and a black dot marks a miss. The location of the dots has nothing to do with where the hits or misses are. The PET just prints all the white dots, followed by all the black dots. If you get no dots at all, none of the colors in your guess are in the pattern.

Finally, if you’re feeling extra sharp, you may put a limit on the number of guesses allowed on a pattern. Exceed the limit and the PET tells you the answer, and marks that game as a loss for you. If you don’t want a limit, just press [RETURN] in answer to the question.

¹ “Master Mind” is a registered trademark of Invicta Plastics (U.S.A.) Ltd.

GAME	1	2	3	4
1	YELLOW	YELLOW	YELLOW	YELLOW
2	BLUE	BLUE	BLUE	BLUE
3	GREEN	GREEN	GREEN	GREEN
4	RED	RED	RED	RED
5	ORANGE	ORANGE	RED	RED
6	VIOLET	VIOLET	VIOLET	RED
7	RED	VIOLET	RED	VIOLET
8	RED	VIOLET	VIOLET	VIOLET
9	VIOLET	VIOLET	RED	VIOLET

THAT WAS THE PATTERN!

CARE TO TRY AGAIN?

```

0 CLR:POKE 59468,12
6 REM MIND COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81RJ
90 PG$="MIND":NM$="7":GOSUB 62000
100 CR$=CHR$(13):PRINT CHR$(142)
105 PRINT "DO YOU WANT A LIMIT ON THE NUMBER OF":PRINT "GUESSES? ";
:GOSUB 60000
108 IF IN$="" THEN PRINT "{down}THERE WILL BE NO LIMIT, THEN.":
FOR I=1 TO 2000:NEXT
110 IN$=LEFT$(IN$,1):IF IN$="N" OR IN$="" THEN LM=1E38:GOTO 200
115 IF IN$<>"Y" THEN PRINT "PLEASE ANSWER 'YES' OR 'NO'.":GOTO 105
120 PRINT "WHAT SHOULD THE LIMIT BE?":GOSUB 60000
130 IF IN$="" THEN PRINT "I'LL ASSUME 10.":LM=10:GOTO 200
140 FOR I=1 TO LEN(IN$):T$=MID$(IN$,I,1)
150 IF T$<"1" OR T$>"9" THEN PRINT "NO.":T$;"S, PLEASE.":GOTO 120
160 NEXT I:LM=VAL(IN$)
200 READ T$:IF T$<>" THEN C=C+1:GOTO 200
205 DATA RED,ORANGE,YELLOW,GREEN,BLUE,VIOLET,
210 N=4:CZ=6:B$="":FOR I=1 TO CZ:B$=B$+" ":L$=L$+"{@" :NEXT I
220 DIM P(N),G(N),PS(N),C$(C)
230 RESTORE:FOR I=1 TO C:READ C$:C$(I)=LEFT$(C$+B$,CZ):NEXT I:
C$(0)=B$
240 H$="{off}":M$="{off}":FOR I=1 TO N:H$=H$+"{Q}":M$=M$+"{W}":
NEXT I
500 TB=6:DEF FN T(N)=(CZ+1)*(N-1)+TB
1000 FOR I=1 TO N:P(I)=INT(RND(1)*C)+1:NEXT I
1010 PRINT "{clr}";:FOR I=1 TO N:PRINT TAB(FN T(I)+(CZ/2)-1);I;:
NEXT I
1020 PRINT :FOR I=1 TO N:PRINT TAB(FN T(I));L$;:NEXT I
1030 GM=GM+1:TN=0:PRINT "{home}GAME":PRINT GM
1100 IF TN>=LM THEN 1960
1110 TN=TN+1:TT=TT+1:PRINT TAB(TB-3);RIGHT$(STR$(TN),2):T=1:OT=T
1120 FOR I=1 TO N:G(I)=0:NEXT I

```

```

1200 IF OT<>T THEN PRINT "{up}";TAB(FN T(OT));C$(G(OT))
1210 IF T<1 THEN T=N
1220 IF T>N THEN T=1
1230 PRINT "{up}";TAB(FN T(T));"{rvs}";C$(G(T)):OT=T
1300 GET T$:IF T$="" THEN 1300
1310 IF T$="." OR T$="{right}" THEN T=T+1:GOTO 1200
1320 IF T$=CHR$(20) OR T$="{left}" THEN T=T-1:GOTO 1200
1330 IF T$=CHR$(13) THEN 2000
1340 FOR I=1 TO C
1350 :IF T$=MID$("123456789",I,1) OR T$=LEFT$(C$(I),1) THEN 1900
1360 NEXT I
1370 IF T$=">" THEN T2=T+1:GOTO 1800
1380 IF T$="<" THEN T2=T-1:GOTO 1800
1390 IF T$="?" THEN 1950
1400 GOTO 1300
1800 IF T2>N THEN T2=1
1810 IF T2<1 THEN T2=N
1820 I=G(T):G(T)=G(T2):G(T2)=I
1830 OT=T2:GOTO 1200
1900 G(T)=I:T=T+1
1910 GOTO 1200
1950 PRINT "{up}....."
1955 PRINT TAB(TB+6);"THE ANSWER IS:{down}":H=0:GOTO 1990
1960 PRINT TAB(TB);"{down}..SORRY, THE SOLUTION WAS:{down}":H=0:
GOTO 1990
1990 LS=LS+1:TT=TT-TN:GOTO 2100
2000 FOR I=1 TO N:IF G(I)=0 THEN 1300
2010 PS(I)=P(I):NEXT I
2015 PRINT "{up}";TAB(FN T(T));C$(G(T))
2020 H=0:FOR I=1 TO N
2030 :IF PS(I)=G(I) THEN H=H+1:PS(I)=-1:G(I)=-2
2040 NEXT I
2050 M=0:FOR I=1 TO N:FOR J=1 TO N
2060 IF PS(I)=G(J) THEN M=M+1:PS(I)=-1:G(J)=-2
2070 NEXT J:NEXT I
2080 PRINT "{up}";TAB(FN T(N+1));LEFT$(H$,H+1);LEFT$(M$,M+1)
2090 IF H<N THEN 1100
2100 PRINT "{up rvs}";TAB(FN T(1));
2110 FOR I=1 TO N-1
2120 :PRINT C$(P(I));".";:NEXT I
2130 PRINT C$(P(N))
2135 IF H=N THEN PRINT TAB(TB);"{down}...THAT WAS THE PATTERN!"
2140 PRINT "{down}CARE TO TRY AGAIN?";:GOSUB 60000
2150 IN$=LEFT$(IN$,1):IF IN$<>"N" THEN 1000
2160 PRINT "{down}VERY WELL.{2°down}":IF LS<GM THEN 2300
2170 IF LM<40 THEN 2250
2190 PRINT ".YOU DIDN'T GUESS.";:IF GM=1 THEN 2210
2200 IF GM=2 THEN PRINT "EITHER":PRINT "OF.";:GOTO 2205
2201 PRINT "ANY OF"
2205 PRINT "THE";GM;"PATTERNS";:GOTO 2220
2210 PRINT "THE":PRINT "PATTERN";

```

```

2220 PRINT ".I.SELECTED.":GOTO 3000
2250 PRINT "TRY.A.HIGHER.GUESS.LIMIT.THE.NEXT.TIME":
PRINT "YOU.PLAY.THIS.";
2260 GOTO 2190
2300 PRINT "YOU.AVERAGED";INT(TT*10/(GM-LS))/10;
"GUESSES.PER.SOLUTION,"
2310 IF LS>0 THEN PRINT "BUT.MISSED";LS;"OUT.OF";GM;"PATTERNS.":
GOTO 3000
2320 PRINT "WITHOUT.MISSING.":IF GM=1 THEN PRINT "THE.PATTERN.":
GOTO 3000
2330 PRINT "ANY.OF.THE";GM;"PATTERNS."
3000 END
60000 IN$="":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "PRESS.{rvs}RETURN{off}_.TO_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE"
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN PRINT "{clr}":CR$=CHR$(13):GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;".-_.CURSOR.#";NM$;
"._(C).1981....PAGE";PN
60270 GOSUB 60500:RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;".BY.GLEN.FISHER"
62002 PRINT "{down}....CURSOR.#7..COPYRIGHT.(C).1981"
62004 GOSUB 60500
62005 PRINT "{down}GUESS{.}THE{.}COLOR.PATTERNS"
62010 PRINT "{2°down}WANT.DIRECTIONS?_";:GOSUB 60000:PRINT :
IN$=LEFT$(IN$,1)
62020 IF IN$="" OR IN$="N" THEN POKE 59468,12:PRINT "{clr}":RETURN
62030 IF IN$<>"Y" THEN PRINT "PLEASE.ANSWER.YES.OR{.}NO.{4°up}":
GOTO 62010
62100 LP=5:GOSUB 60200
62110 PRINT "MASTERMIND.IS.A.GAME.IN.WHICH.YOU.TRY"
62120 PRINT "TO.GUESS.THE.COLOR.PATTERN.THE.PET.HAS"

```

```

62130 PRINT "CHOSEN..AFTER.EACH.GUESS.YOU.MAKE,.THE"
62140 PRINT "PET.WILL.GIVE.YOU.HINTS.ABOUT.HOW.CLOSE"
62150 PRINT "YOU.CAME.{down}"
62160 PRINT "YOU.WILL.BE.TOLD.HOW.MANY.HITS.AND"
62170 PRINT "MISSES.YOU.GOT.WITH.THAT.GUESS."
62180 PRINT "{down}YOU.SCORE.A.HIT.WHENEVER.A.COLOR.IN"
62190 PRINT "YOUR.GUESS.MATCHES.A.COLOR.IN.THE.PET'S"
62200 PRINT "PATTERN.EXACTLY,.IN.COLOR.AND.POSITION."
62210 PRINT "FOR.EXAMPLE"
62220 PRINT "{down}PATTERN:.RED...GREEN...BLUE...VIOLET"
62240 PRINT "GUESS:...BLUE...GREEN...RED...ORANGE"
62250 PRINT "{down}WOULD.GET.A.HIT.FOR.THE{.}GREEN."
62260 GOSUB 60200
62270 PRINT "YOU.GET.A.MISS.WHENEVER.A.COLOR.IN.YOUR"
62280 PRINT "GUESS.MATCHES.A.COLOR.IN.THE.PATTERN,"
62290 PRINT "BUT.ISN'T.IN.THE.RIGHT.POSITION..FOR"
62300 PRINT "EXAMPLE"
62310 PRINT "{down}PATTERN:.GREEN...GREEN...BLUE....RED"
62330 PRINT "GUESS:...BLUE...ORANGE...ORANGE...VIOLET"
62340 PRINT "{down}WOULD.GET.A.MISS.ON.THE.BLUE."
62350 PRINT "{down}ANY.GIVEN.COLOR.IN.YOUR.GUESS.CAN.MATCH"
62360 PRINT "ONLY.ONE.COLOR.IN.THE.PATTERN,.EVEN.IF"
62370 PRINT "THERE.ARE.DUPLICATES..FOR.EXAMPLE"
62380 PRINT "{down}PATTERN:.RED..RED.....ORANGE...ORANGE"
62400 PRINT "GUESS:...RED...ORANGE...BLUE....GREEN"
62410 PRINT "{down}WOULD.GET.ONE.HIT.(FOR.RED).AND{.}"
62420 PRINT "ONE.MISS{.}(FOR{.}ORANGE). "
62430 GOSUB 60200
62440 PRINT "SIMILARLY,.ANY.GIVEN.COLOR{.}IN.THE"
62450 PRINT "PATTERN.CAN.MATCH.ONLY.ONE.COLOR.IN"
62460 PRINT "YOUR.GUESS..FOR.EXAMPLE"
62470 PRINT "{down}PATTERN:.RED..ORANGE...BLUE....GREEN"
62480 PRINT "GUESS:...RED..RED.....ORANGE...ORANGE"
62490 PRINT "{down}WOULD.GET.ONE.HIT{.}(FOR{.}RED),.AND"
62500 PRINT "ONE.MISS.(FOR{.}ORANGE). "
62510 PRINT "{2°down}TO{.}ENTER.YOUR.GUESS,.YOU.TYPE.IN.THE"
62520 PRINT "COLORS.YOU.WANT{.}(USING.EITHER.THE.FIRST"
62530 PRINT "LETTERS.OF.THE.WORDS{.}RED,.ORANGE,"
62540 PRINT "YELLOW,.GREEN,.BLUE,.AND{.}VIOLET,.OR.BY"
62560 PRINT "TYPING.A.DIGIT.FROM.1.TO{.}6.(THE.COLORS"
62570 PRINT "ARE.NUMBERED.IN.THE.ORDER{.}RED.THROUGH"
62580 PRINT "VIOLET)....THE.POSITION.THE{.}COLOR.WILL"
62590 PRINT "GO.INTO.IS.MARKED.WITH{. rvs}BLACK-ON-WHITE{off}."
62600 GOSUB 60200
62610 PRINT "YOU.MAY.MODIFY.YOUR.GUESS.(TO.CORRECT.A"
62620 PRINT "MISTYPE,.OR.CHANGE.YOUR.MIND).BY.USING"
62630 PRINT "THE{.}DEL,.LEFT,.RIGHT,.AND{.}SPACE{.}KEYS."
62640 PRINT "DEL.AND{.}LEFT.MOVE{.}THE{. rvs}MARKER{off}.TO.THE"
62650 PRINT "LEFT,.AND.RIGHT.AND{.}SPACE.MOVE.IT.TO"
62660 PRINT "RIGHT..IF.THE{. rvs}MARKER{off}.GOES.OFF.EITHER"
62670 PRINT "END,.IT.WILL.REAPPEAR.AT.THE.OTHER."

```

```

62680 PRINT "IN.ADDITION,,THE{.}'<'AND{.}'>'KEYS"
62690 PRINT "WILL.SWITCH.COLORS.. '<'WILL.SWITCH.THE"
62700 PRINT "COLOR.UNDER.THE{. rvs}MARKER{off}.WITH.THE.COLOR"
62710 PRINT "TO.ITS.LEFT,,AND{.}'>'WITH.THE.COLOR.TO"
62720 PRINT "THE.RIGHT.OF.THE{. rvs}MARKER{off}."
62730 PRINT "{down}ALSO,,TO.GIVE.UP.AND.SEE.THE.PATTERN,"
62740 PRINT "YOU.CAN.PRESS.THE.'?'KEY."
62750 PRINT "{down}WHEN.YOUR.GUESS.IS.ENTERED.TO.YOUR"
62760 PRINT "SATISFACTION,,PRESS.RETURN.TO.FIND.OUT"
62770 PRINT "HOW.MANY.HITS.AND.MISSES.YOU.GOT."
62780 GOSUB 60200
62790 PRINT "THE.PET.WILL.SHOW.YOUR.HITS.AND.MISSES"
62800 PRINT "BY.PRINTING,,BESIDE.YOUR.GUESS,,A.SOLID"
62810 PRINT "DOT.FOR.EACH.HIT,,AND.A.HOLLOW.DOT.FOR"
62820 PRINT "EACH.MISS..FOR.EXAMPLE,,2.HITS.AND.1"
62830 PRINT "MISS.WOULD.LOOK.LIKE.THIS:"
62840 PRINT "{down}...{QQW}"
62860 PRINT "{down}WHEN.YOU'RE.ALL.DONE.GUESSING.PATTERNS,"
62870 PRINT "THE.PET.WILL.TELL.YOU.HOW.WELL.YOU.DID"
62880 PRINT "OVER.ALL.THE.PATTERNS."
62890 PRINT "{3°down}GOOD.LUCK.{2°down}"
63000 GOSUB 60200

```



PEG

By Julia Hallford

Here's a simple one. With two sets of four pegs, in a board with nine holes in a straight line, all you have to do is switch the two sets of pegs. Seems easy, huh? Well, there are a few rules we haven't told you yet.

The pegs are set up in the board with four to the left (marked ">") and four to the right (marked "<"), with a hole between them. On any move, you may move a peg one place left or right (according to the arrow on top); ">" pegs can move only to the right, and a "<" peg only to the left, and must move into the hole. Also you may jump over a peg going in the other direction, if you can land in the hole. These are your only choices: move one place into the hole

or jump over an opposite-way peg into the hole. Other than that, you may move any peg you like at any time.

To move, tell the PET which peg you want to move (all are numbered). Since it must move into the hole, and there's only one hole, the PET knows where its going and you don't need to tell it that. The PET won't allow any moves which violate the rules. When you've got the pegs switched, with the hole once again in the middle, you're done. The PET will tell you how many moves you took to do it (the minimum is 24). If you get stuck, or give up, move peg zero to quit.

Incidentally, the PET knows how to solve the puzzle, and will gladly show you if you ask.

```
0 CLR:POKE 59468,12:PK=12
6 REM PEG COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="PEG":NM$="11":GOSUB 62000
100 GOSUB 60400
105 BB$="....."
110 PRINT "{down}WANT_TO_SEE_HOW_TO_SOLVE_THE_PUZZLE? ";
120 GOSUB 60000
130 PRINT "{clr}";:GOSUB 1000
140 IF LEFT$(IN$,1)="Y" THEN 1200
150 M=0:X=5
160 PRINT "{home}";BB$
180 M=M+1:DL=0:PRINT "{home}MOVE.#"M:Y=X
190 GOSUB 800:GOSUB 700:DL=1000
200 PRINT TAB(6);"FROM? ";:GOSUB 60000
201 IF IN$="" THEN PRINT "{up}";:GOTO 200
202 IF ASC(IN$)=32 THEN IN$=MID$(IN$,2):GOTO 201
205 X=VAL(IN$)
210 IF X=0 THEN 470
220 IF X<1 OR X>9 THEN PRINT "TYPE_A_NUMBER_FROM_1_TO_9,_PLEASE.":
GOTO 190
230 IF ABS(X-Y)>2 THEN 450
240 IF P(X)=32 GOTO 450
260 IF P(X)=62 AND X>Y THEN 450
270 IF P(X)=60 AND X<Y THEN 450
280 P(Y)=P(X):P(X)=32
290 JX=CRT+WD*8+9+2*X:JY=CRT+WD*8+9+2*Y
320 POKE JY,P(Y):POKE JX,32
```

[illegible]


```

1620 GOSUB 1000:GOTO 150
2000 DATA 62,62,62,62,32,60,60,60,60
2030 DATA 6,5,4,6,3,4,5,3,7,5,8,7,6,8,4,6,2,4,1,2,3,1,5,3
2040 DATA 7,5,9,7,8,9,6,8,4,6,2,4,3,2,5,3,7,5,6,7,4,6,5,4
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN POKE 59468,PK:CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_-_{CURSOR}_#";NM$;
      "_{(C)}_1981.....PAGE";PN
60270 GOSUB 60500:RETURN
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "@@@";:NEXT:PRINT :RETURN
62000 PRINT "{clr down}";TAB(15-LEN(PG$));PG$;"_BY_JULIA_HALLFORD"
62010 PRINT "{down}.....CURSOR_#11..COPYRIGHT_(C)_1981"
62020 GOSUB 60500
62025 PRINT "{down}REVERSE_TWO_SETS_OF_PEGS_ON_A_BOARD"
62030 PRINT "{2°down}WANT_INSTRUCTIONS?_";:GOSUB 60000
62040 IF LEFT$(IN$,1)<>"Y" THEN PRINT "{clr}";:POKE 59468,PK:CLR:
      GOTO 100
62100 LP=2:GOSUB 60200
62400 PRINT "THIS_GAME_IS_PLAYED_WITH_TWO_SETS_OF"
62410 PRINT "{down}FOUR_PEGS_REPRESENTED_BY_'<{'right}AND'>'."
62420 PRINT "{down}THE_BOARD_STARTS_OUT_WITH_A_SET_OF_PEGS"
62430 PRINT "{down}AT_EACH_{ }END,_WITH_A_SPACE_BETWEEN_THEM."
62440 PRINT "{2°down}THE_OBJECT_OF_THE_GAME_IS_TO_SWITCH_THE"
62450 PRINT "{down}TWO_SETS_OF_{ }PEGS..A_PEG_MAY_EITHER_MOVE"
62460 PRINT "{down}INTO_THE_SPACE,_IF_THE_SPACE_IS_NEXT_TO"
62470 PRINT "{down}THE_PEG,_OR_JUMP_OVER_ONE_PEG_INTO_THE"
62480 PRINT "{down}SPACE.._'>'_PEGS_MOVE_TO_THE_RIGHT,_AND"

```

```
62490 PRINT "{down}'<'_PEGS_TO_THE_LEFT."
62495 GOSUB 60200
62500 PRINT "TO_MOVE,_TYPE_THE_NUMBER_OF_THE_SPACE"
62510 PRINT "{down}FROM_WHICH_YOU_WANT_THE_PEG_MOVED,_AND"
62520 PRINT "{down}HIT_{rvs}RETURN{off}._IF_YOU_REACH_A_POSITION"
62530 PRINT "{down}IN_WHICH_YOU_HAVE_NO_LEGAL_MOVE,_ENTER"
62545 PRINT "{down}A_'0'_TO_RESIGN."
62550 PRINT "{2°down}THE_PUZZLE_CAN_BE_SOLVED_IN{ }24_MOVES."
62555 PRINT "{down}SEE_HOW_WELL_YOU_CAN_DO."
62560 GOSUB 60200
```



By Glen Fisher

You are playing against yourself in LEAP, trying to eliminate all but one of the pegs on the board. When the game starts, you have a 4×8 board filled with holes which contain 31 pegs. The rows are numbered and the columns are lettered, so any given hole is described by a letter/number combination (for example, D2 or 2D). The PET will ask you where you want the space to be. You may either name the place for the hole, or just press [RETURN] and let the PET pick a spot at random.

On each move, you must remove a peg by leaping over it with another. Pegs can only leap into holes. Each peg that you jump over is removed, and a hole is left in its place. The *only* kind of move you can make is a leap; just moving the peg is illegal. The game is over when you can't make any more leaps. To achieve a perfect game, you must leave only one peg behind. (Two isn't hard, but one is very difficult.)

To leap over a peg, type the positions of the peg you want to move when the PET gives you the “From ?” prompt. Then type the position of the hole you want to leap into when the PET gives you the “To ?” prompt. For the “To ?” prompt, you can give just the part of the position that changes. (For example, with a move from C4 to

C2, you could respond to “To ?” with just a “2”, since the “C” stays the same.) If you decide you want to move another peg, answer the “To ?” prompt with a [left arrow], and you’ll get the “From ?” prompt back again. You can move any direction except diagonally. If you want to resign, press [R].

[illegible][illegible]

```

1050 PRINT TAB(HT); "{-@l@l@l@l@l@l@l@=}"
1110 FOR I=1 TO 4:FOR J=1 TO 8:B(I,J)=1:NEXT J:NEXT I
1200 FOR C=1 TO 8:FOR R=1 TO 4
1210 GOSUB 9200
1220 NEXT R,C
1300 IF HC>0 THEN R=HR:C=HC:GOTO 1330
1310 H=-1:V=9:GOSUB 9000:PRINT "WHERE'S THE HOLE?...{2°left}";:
    GOSUB 60000:GOSUB 9100
1315 IF IN$="" THEN R=INT(RND(1)*3)+1:C=INT(RND(1)*8)+1:ER=0
1320 IF ER THEN 1310
1325 HR=R:HC=C
1330 B(R,C)=0:GOSUB 9200
1340 H=-1:V=9:GOSUB 9000:PRINT "....."
1900 MV=0:TIS="000000"
2000 H=5:V=-3:GOSUB 9000:PRINT "{rvs}MOVE";MV+1:FR=0:FC=0
2005 H=0:V=9:GOSUB 9000:PRINT "FROM?...{2°left}";:GOSUB 60000:
    GOSUB 9100
2010 ON ER+1 GOTO 2020,2000,2950,2800
2020 FR=R:FC=C
2030 IF B(FR,FC)=0 THEN 2000
2040 GOSUB 9250
2050 H=0:V=9:GOSUB 9000:PRINT "TO?.....{4°left}";:GOSUB 60000:
    GOSUB 9100
2055 ON ER+1 GOTO 2060,2050,2950,2800
2060 TR=R:TC=C
2070 IF B(TR,TC)<>0 THEN 2050
2100 IF FR<>TR AND FC<>TC THEN 2900
2110 IF ABS(FR-TR)<>2 AND FC=TC THEN 2900
2120 IF ABS(FC-TC)<>2 AND FR=TR THEN 2900
2130 MR=(FR+TR)/2:MC=(FC+TC)/2
2140 IF B(MR,MC)=0 THEN 2900
2200 B(TR,TC)=B(FR,FC):B(FR,FC)=0
2210 B(MR,MC)=0
2220 R=FR:C=FC:GOSUB 9200
2230 R=TR:C=TC:GOSUB 9200
2240 R=MR:C=MC:GOSUB 9200
2250 MV=MV+1
2260 H=2:V=13:GOSUB 9000:PRINT "TIME_";:ET$=TIS
2270 PRINT LEFT$(ET$,2);": ";MID$(ET$,3,2);": ";RIGHT$(ET$,2)
2300 FOR I=1 TO 4:FOR J=1 TO 8:T=SGN(B(I,J))
2310 IF I<3 THEN IF T<>SGN(B(I+2,J)) AND B(I+1,J)<>0 THEN 2000
2320 IF J<7 THEN IF T<>SGN(B(I,J+2)) AND B(I,J+1)<>0 THEN 2000
2330 NEXT J,I
2400 H=0:V=11:GOSUB 9000:PRINT "YOU HAVE NO MOVE."
2410 PG=31-MV:IF PG<BP THEN BP=PG:BT$=ET$
2420 IF PG=BP AND ET$<BT$ THEN BT$=ET$
2500 H=0:V=15:GOSUB 9000:PRINT "TRY AGAIN?_";:GOSUB 60000
2510 IF LEFT$(IN$,1)<>"N" THEN 1000
2520 PRINT "{clr 2°down}YOUR BEST GAME WAS:.....{down}"
2530 PRINT BP;"PEGS LEFT IN_";
2540 IF LEFT$(ET$,2)<>"00" THEN PRINT LEFT$(ET$,2);": ";

```

```

2550 PRINT MID$(BT$,3,2);":";RIGHT$(BT$,2);"."
2590 END
2800 ET$=TI$:GOTO 2410
2900 IF ER<0 THEN 2950
2905 H=4:V=11:GOSUB 9000:PRINT "{rvs}BAD_MOVE."
2910 TM=TI+90
2920 IF TI<TM THEN 2920
2930 GOSUB 9000:PRINT "....."
2950 R=FR:C=FC:GOSUB 9200
2960 GOTO 2000
8999 END
9000 PRINT "{home}":POKE QL,V+VT:PRINT "{up}";TAB(HT+H);:RETURN
9100 ER=1:IF IN$="<" THEN ER=2
9101 IF IN$="R" THEN ER=3
9102 IF LEN(IN$)=1 AND FR<>0 THEN 9170
9105 IF LEN(IN$)<>2 THEN RETURN
9110 R$=LEFT$(IN$,1):C$=RIGHT$(IN$,1)
9120 IF R$>="A" AND R$<="H" THEN 9150
9130 IF R$<"1" OR R$>"4" OR C$<"A" OR C$>"H" THEN RETURN
9140 R=ASC(R$)-48:C=ASC(C$)-64:ER=0:RETURN
9150 IF C$<"1" OR C$>"4" THEN RETURN
9160 R=ASC(C$)-48:C=ASC(R$)-64:ER=0:RETURN
9170 IF IN$>="A" AND IN$<="H" THEN C=ASC(IN$)-64:R=FR:ER=0:RETURN
9180 IF IN$>="1" AND IN$<="4" THEN R=ASC(IN$)-48:C=FC:ER=0:RETURN
9190 RETURN
9200 H=2*C-1:V=2*R-1:GOSUB 9000:PRINT MID$(DT$,1+2*B(R,C),2):RETURN
9250 B=B(R,C):B(R,C)=B+3:GOSUB 9200:B(R,C)=B:RETURN
9999 END
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>2 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240

```

```

60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_._CURSOR.#";NM$;
      "._.(C)._1981._.PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000)=0 THEN RETURN
60420 QL=216:POKE 1022,128
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_._BY_GLEN_FISHER"
62010 PRINT "{down}._.CURSOR.#13._.COPYRIGHT._.(C)._1981"
62020 GOSUB 60500
62030 PRINT "{down}CLEAR_THE_BOARD_OF_ALL_THE_PEGS"
62080 PRINT "{2°down}WANT_DIRECTIONS?._";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=2:GOSUB 60200
62110 PRINT "YOU_BEGIN_WITH_A_4_BY_8_BOARD_WITH_ALL"
62120 PRINT "{down}SQUARES_FILLED_WITH_WHITE_PEGS._YOU"
62130 PRINT "{down}FIRST_REMOVE_ONE_PEG._(IF_YOU_PRESS"
62140 PRINT "{down rvs}RETURN{off},_A_RANDOM_PEG_IS_REMOVED.)"
62150 PRINT "{2°down}AFTER_THE_STARTING_'HOLE'_IS_SELECTED,"
62160 PRINT "{down}THE_OBJECT_OF_THE_GAME_IS_TO_REMOVE_PES{left
      ins}G"
62170 PRINT "{down}BY_JUMPING_OTHER_PEGS_OVER_THEM._NOTE"
62180 PRINT "{down}THAT_YOU_MAY_ONLY_JUMP_UP_AND_DOWN,_OR"
62190 PRINT "{down}LEFT_AND_RIGHT,_BUT_{rvs}NOT{off}_DIAGONALLY!"
62195 GOSUB 60200
62200 PRINT "THE_GAME_ENDS_WHEN_YOU_HAVE_ONLY_ONE_PG{left ins}E"
62210 PRINT "{down}LEFT,_OR_WHEN_YOU_HAVE_NO_REMAINING"
62220 PRINT "{down}LEGAL_MOVES._YOU_CAN_'RESIGN'_BY_PRESS-"
62230 PRINT "{down}ING_'R'_FOR_YOUR_'FROM'_MOVE."
62240 PRINT "{2°down}IF_YOU_CHANGE_YOUR_MIND_ABOUT_WHICH_PEG"
62250 PRINT "{down}YOU_WANT_TO_MOVE,_ENTER_A_'<'_FOR_YOUR"
62260 PRINT "{down}'TO'_MOVE,_AND_YOU'LL_BE_ASKED_AGAIN"
62270 PRINT "{down}WHICH_PEG_YOU_WISH_TO_MOVE."
62280 GOSUB 60200

```



By David Plotton

2 MOVES

```

300 NEXT J:N=0:PRINT "{3°down}":NEXT I:POKE QL,22:
    PRINT "{up}"TAB(13)N;"MOVES"
310 GET Z$:IF Z$="" THEN 310
315 IF Z$="Q" THEN PRINT "{clr}":END
320 IF Z$<"2" OR Z$>"8" THEN 310
330 IF VAL(Z$) AND 1 THEN 310
340 ON VAL(Z$)/2 GOTO 410,390,370,350
350 T=BS+4:IF T>16 THEN 310
360 GOTO 420
370 T=BS-1:IF (3 AND T)=0 THEN 310
380 GOTO 420
390 T=BS+1:IF (3 AND BS)=0 THEN 310
400 GOTO 420
410 T=BS-4:IF T<1 THEN 310
420 B(BS)=B(T):B(T)=16
430 I=INT((BS-1)/4):J=BS-I*4
440 POKE QL,I*4+5:PRINT "{up}"TAB(TB+J*5-3)RIGHT$(STR$(B(BS)),2)
450 I=INT((T-1)/4):J=T-I*4
460 POKE QL,I*4+5:PRINT "{up}"TAB(TB+J*5-3)"_._"
470 BS=T:N=N+1:POKE QL,22:PRINT "{up}"TAB(13)STR$(N);"_MOVE";:
    IF N>1 THEN PRINT "S";
475 PRINT "_":IF BS<16 THEN 310
480 FOR I=1 TO 16:IF B(I)<>I THEN 310
490 NEXT I:POKE QL,21:PRINT "{up}"TAB(14)"SOLVED_ IN{2°down}"
600 PRINT "DO_YOU_WANT_TO_PLAY_AGAIN?_";:GOSUB 60000:IF LEFT$(IN$,
    1)<>"N" THEN 120
700 END
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_&{&}",ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000
60140 GOTO 60010
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000) THEN QL=216
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_BY_DAVID_PLATTON"
62010 PRINT "{down}_.CURSOR_#15_.COPYRIGHT_(C)_1981"

```



```
62020 GOSUB 60500
62030 PRINT "{down}THE.FIFTEEN.PUZZLE"
62040 PRINT "{2°down}PRESS.{rvs}RETURN{off}.TO.BEGIN"
62050 GET T$:IF T$="" THEN 62050
62060 GOTO 60300
```



RATRUN

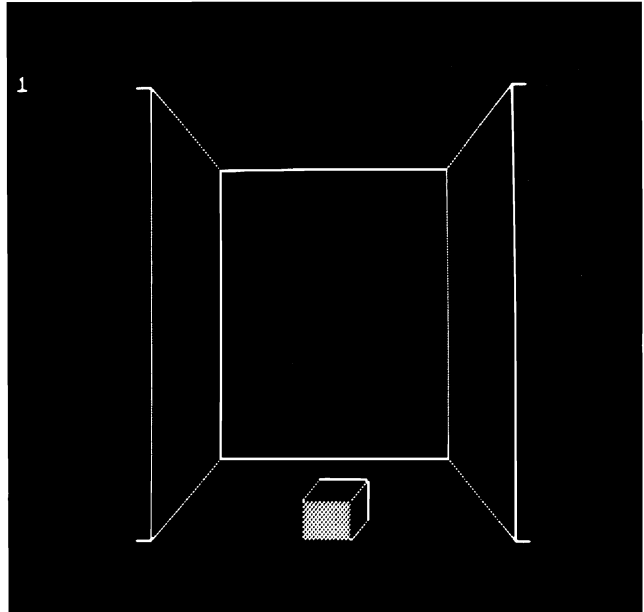
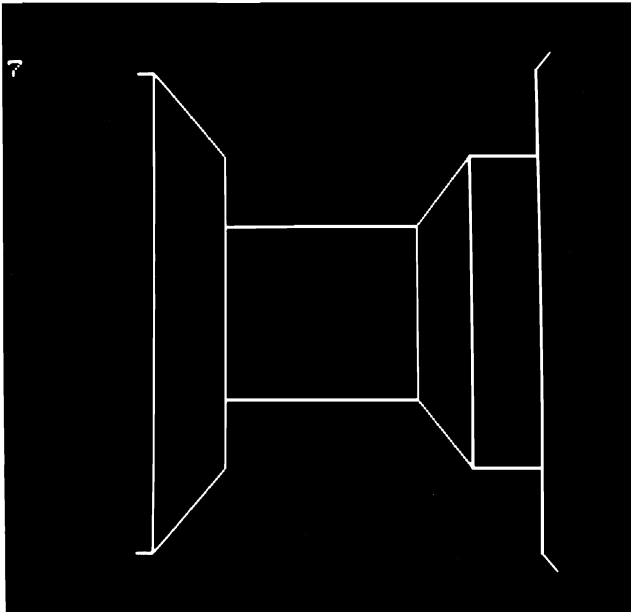
By Chris Nadovich

Yes indeed, here is yet another maze-making program. However, this one is different. In this program, you are *inside* the maze, not above it. The goal is a piece of cheese (this is a *rat* run, after all) somewhere in the maze. Since the rat can probably smell the cheese better than you can, we'll give you one additional clue. In a corner of the screen, you're told how far it is to the cheese. That's not as much help as you might think; the quoted distance is "as the flea jumps," ignoring the walls of the maze. Thus, it tells how far away you are from the cheese, not how far you must travel to reach it.

To navigate the maze, you may press the [8] key to move forward. You may also press the [4]

key to turn to the left, [6] to turn to the right, or [2] to turn around backward. The [4], [6], and [2] keys do *not* move you, they just turn you. Only the [8] key moves you.

Finally, if you despair of ever finding the cheese, press [?]. You'll be given a map of the maze. Where you started will be marked with a big white square, your travels will be shown by periods in the squares you've occupied, and the cheese will be marked with a grey square. Do not give up easily. Once you see the map, you'll never be allowed into *that* maze again. Other mazes are fine, (and you'll be offered one), but never that one. **WARNING:** If you get nervous in elevators or closed spaces, do not play this game alone at night.

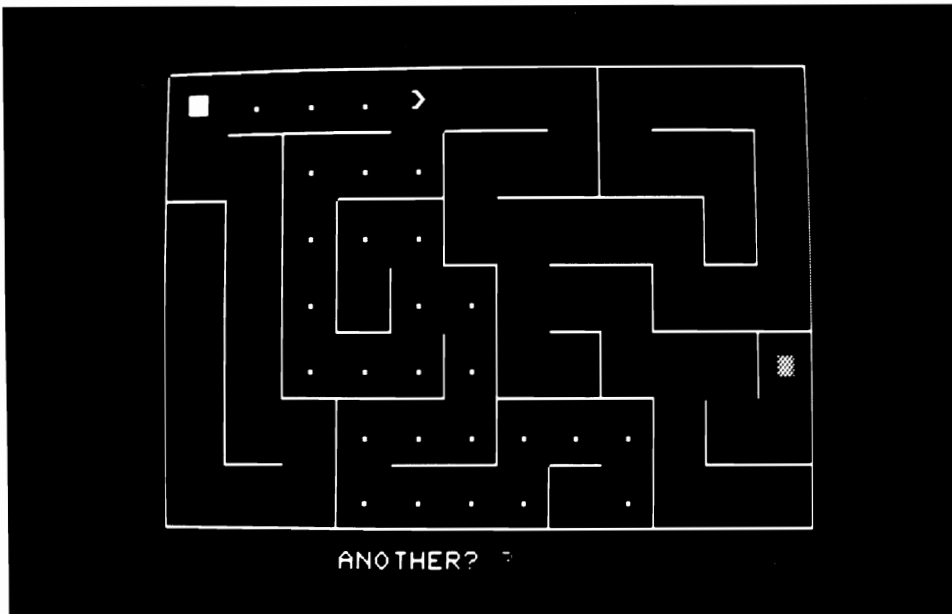


```
0 CLR:POKE 59468,12
6 REM  RATRUN COPYRIGHT (C) 1981 THE CODE WORKS
10 REM  AS OF 10/8/81 RJ
90 PG$="RATRUN":NM$="13":GOSUB 62000
100 PRINT CHR$(142):H=12:V=7:H1=H+1:W=634:V1=826
110 I=RND(-TI)
130 DIM XL(4),N2(4),YL(4),XR(4)
140 FOR I=0 TO 4:READ XL(I),N2(I),YL(I),XR(I):NEXT I
150 DATA 8,20,-4,31,8,12,1,27,12,6,5,24,15,2,8,22,17,0,10,21
200 CX=INT(RND(1)*H)+1:CY=INT(RND(1)*V)+1
210 C=CX:R=CY:S=0
```

```

220 FOR I=0 TO (V+1)*H1:POKE W+I,0:POKE V1+I,0:NEXT
230 PRINT "{clr 2°down}SETTING UP MAZE...";
300 DA=0:N=0:AV=V1+C+R*H1:AW=W+C+R*H1:POKE AW,1:S=S+1:
    IF S>=H*V THEN 900
320 PRINT MID$("{rvs off}",(S AND 1)+1,1);"{V left}";
410 IF C>1 AND PEEK(AW-1)=0 THEN DA=DA+2:N=N+1
420 IF C<H AND PEEK(AW+1)=0 THEN DA=DA+1:N=N+1
430 IF R>1 AND PEEK(AW-H1)=0 THEN DA=DA+8:N=N+1
440 IF (R<V AND PEEK(AW+H1)=0) THEN DA=DA+4:N=N+1
450 N=INT(RND(1)*N)+1
460 ON DA+1 GOTO 500,850,800,600,750,610,620,630,700,640,650,660,
    670,680,690,695
500 S=S-1
510 C=C+1:IF C>H THEN C=1:R=R+1:IF R>V THEN R=1
520 ON PEEK(W+C+R*H1)+1 GOTO 510,300
600 ON N GOTO 800,850
610 ON N GOTO 750,850
620 ON N GOTO 750,800
630 ON N GOTO 750,800,850
640 ON N GOTO 700,850
650 ON N GOTO 700,800
660 ON N GOTO 700,800,850

```



```

670 ON N GOTO 700,750
680 ON N GOTO 700,750,850
690 ON N GOTO 700,750,800
695 ON N GOTO 700,750,800,850
700 R=R-1:AV=AV-H1:POKE AV,PEEK(AV) OR 1:GOTO 300
750 POKE AV,PEEK(AV) OR 1:R=R+1:AV=AV+H1:GOTO 300
800 C=C-1:AV=AV-1:POKE AV,PEEK(AV) OR 2:GOTO 300
850 POKE AV,PEEK(AV) OR 2:C=C+1:AV=AV+1:GOTO 300
900 I=RND(1)>.5:J=RND(1):K=RND(1)>.5
920 IF I THEN X=INT(J*H)+1:Y=V+(V-1)*K

```

```

930 IF I=0 THEN Y=INT(J*V)+1:X=H+(H-1)*K
931 IF (CX-X)2+(CY-Y)2<(H2+V2)/9 THEN 900
950 SX=X:SY=Y
1001 PRINT "{clr}"
1010 FOR I=W TO W+(V+1)*H1:POKE I,0:NEXT
1020 D=INT(RND(1)*4)
1030 T4=TI:POKE (W+X+Y*H1),1
1045 IF F=0 THEN GOSUB 3000:PRINT "{home}";
      INT(SQR((X-CX)2+(Y-CY)2)*10+.5)/10
1049 IF CX=X AND CY=Y THEN 6000
1050 GET Q$:IF Q$=""GOTO 1050
1060 IF Q$="Q" OR Q$="?" THEN K=TI:GOTO 6100
1070 I=VAL(Q$):IF I=0 THEN 1050
1075 IF I=2 THEN D=D+2
1080 IF I=4 THEN D=D-1
1090 IF I=6 THEN D=D+1
1100 IF D<0 THEN D=D+4
1110 IF D>3 THEN D=D-4
1120 IF I<>8 THEN 1045
1200 AV=V1+X+Y*H1:AW=W+X+Y*H1
1201 ON D GOTO 1260,1310,1360
1210 IF Y>1 AND (PEEK(AV-H1) AND 1) THEN Y=Y-1:POKE AW-H1,1:
      GOTO 1045
1220 GOTO 1400
1260 IF X<H AND (PEEK(AV) AND 2) THEN X=X+1:POKE AW+1,1:GOTO 1045
1290 GOTO 1400
1310 IF Y<V AND (PEEK(AV) AND 1) THEN Y=Y+1:POKE AW+H1,1:GOTO 1045
1340 GOTO 1400
1360 IF X>1 AND (PEEK(AV-1) AND 2) THEN X=X-1:POKE AW-1,1:GOTO 1045
1400 DN=23:GOSUB 7000
1410 PRINT TAB(16);"{$$$$$$$ 7°left down rvs}NO_DOOR{up}":FOR I=1 TO
      1000:NEXT
1420 PRINT TAB(16);"{up}.....{7°left down}.....{up}":GOTO 1049
2000 PRINT "{clr}.";
2020 FOR I=1 TO H:PRINT "$$$";:NEXT:PRINT "{2°down}"
2100 FOR J=1 TO V:PRINT "' up left ' up left ' 2°down";:FOR I=1 TO
      H
2125 K=PEEK(V1+I+J*H1)
2130 PRINT MID$("{$:}...{'$$$}...",1+3*K,3);
2140 IF K<2 THEN PRINT "{up left ' up left ' 2°down}";
2200 PRINT "{up left}";
2210 IF PEEK(W+I+J*H1)=1 THEN PRINT "{left}.";
2250 IF I=SX AND J=SY THEN PRINT "{left rvs}._{off}";
2260 IF I=CX AND J=CY THEN PRINT "{left &}";
2270 IF I=X AND J=Y THEN PRINT "{left}";MID$("↑>V<",D+1,1);
2280 PRINT "{right down}";
2290 NEXT:PRINT "{2°down}":NEXT:RETURN
3000 S=0:PRINT "{clr}"
3005 AV=V1+X+Y*H1
3010 ON D GOTO 3200,3400,3600
3025 IF SY=Y-S AND SX=X THEN GOSUB 5100

```

```

3026 IF CY=Y-S AND CX=X THEN GOSUB 5000
3027 K=PEEK(AV-(S+1)*H1):IF K=0 OR K=2 GOTO 3100
3030 IF PEEK(AV-S*H1) AND 2 THEN GOSUB 4000:GOTO 3040
3035 GOSUB 4200
3040 IF PEEK(AV-1-S*H1) AND 2 THEN GOSUB 4500:GOTO 3050
3045 GOSUB 4700
3050 S=S+1:IF S>4 GOTO 3900
3060 GOTO 3025
3100 IF PEEK(AV-S*H1) AND 2 THEN GOSUB 4300:GOTO 3110
3105 GOSUB 4800
3110 IF PEEK(AV-1-S*H1) AND 2 THEN GOSUB 4400:GOTO 3120
3115 GOSUB 4900
3120 RETURN
3200 IF CY=Y AND CX=X+S THEN GOSUB 5000
3210 IF SY=Y AND SX=X+S THEN GOSUB 5100
3226 K=PEEK(AV+S):IF K=0 OR K=1 GOTO 3300
3230 IF PEEK(AV+S)=2 THEN GOSUB 4200:GOTO 3240
3235 GOSUB 4000
3240 IF PEEK(AV+S-H1) AND 1 THEN GOSUB 4500:GOTO 3250
3245 GOSUB 4700
3250 S=S+1:IF S>4 GOTO 3900
3260 GOTO 3200
3300 IF PEEK(AV+S)=0 THEN GOSUB 4800:GOTO 3310
3305 GOSUB 4300
3310 IF PEEK(AV+S-H1) AND 1 THEN GOSUB 4400:GOTO 3320
3315 GOSUB 4900
3320 RETURN
3400 IF CY=Y+S AND CX=X THEN GOSUB 5000
3410 IF SY=Y+S AND SX=X THEN GOSUB 5100
3426 K=PEEK(AV+S*H1):IF K=0 OR K=2 THEN GOTO 3500
3430 IF PEEK(AV-1+S*H1) AND 2 THEN GOSUB 4000:GOTO 3440
3435 GOSUB 4200
3440 IF PEEK(AV+S*H1)=1 THEN GOSUB 4700:GOTO 3450
3445 GOSUB 4500
3450 S=S+1:IF S>4 GOTO 3900
3460 GOTO 3400
3500 IF PEEK(AV-1+S*H1) AND 2 THEN GOSUB 4300:GOTO 3510
3505 GOSUB 4800
3510 IF PEEK(AV+S*H1)=0 THEN GOSUB 4900:GOTO 3520
3515 GOSUB 4400
3520 RETURN
3600 IF CX=X-S AND CY=Y THEN GOSUB 5000
3610 IF SX=X-S AND SY=Y THEN GOSUB 5100
3625 K=PEEK(AV-(S+1)):IF K=0 OR K=1 THEN GOTO 3700
3630 IF PEEK(AV-S-H1) AND 1 THEN GOSUB 4000:GOTO 3640
3635 GOSUB 4200
3640 IF PEEK(AV-S) AND 1 THEN GOSUB 4500:GOTO 3650
3645 GOSUB 4700
3650 S=S+1:IF S>4 THEN 3900
3660 GOTO 3600
3700 IF PEEK(AV-S-H1) AND 1 THEN GOSUB 4300:GOTO 3710

```

```

3705 GOSUB 4800
3710 IF PEEK(AV-S) AND 1 THEN GOSUB 4400:GOTO 3720
3715 GOSUB 4900
3720 RETURN
3900 DN=11:GOSUB 7000:PRINT TAB(19){MN down 2°left NM home}:RETURN
4000 DN=YL(S):GOSUB 7000
4015 IF S=0 THEN PRINT TAB(XR(S));"${}$"
4020 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XR(S)+5-S){%}":
NEXT
4045 IF S>0 THEN PRINT TAB(XR(S))MID$("${$$$%}",S)
4070 IF S=4 THEN PRINT TAB(XR(S)){%% down 2°left %%}
4071 Q$="":IF S>0 THEN Q$=MID$("{3°right %}",S)
4072 IF S<4 THEN FOR I=0 TO N2(S)+1:PRINT TAB(XR(S)){%}"Q$:NEXT
4075 IF S>0 THEN PRINT TAB(XR(S))MID$("#{###%}",S)
4095 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XR(S)+5-S){%}":
NEXT
4096 IF S=0 THEN PRINT TAB(XR(S)){#}
4098 PRINT "{home}"
4099 RETURN
4200 DN=YL(S):GOSUB 7000
4215 IF S=0 THEN PRINT TAB(XR(S)){N}:GOTO 4230
4220 FOR I=1 TO 5-S:PRINT TAB(XR(S)+5-S-I){N}:NEXT
4230 DN=N2(S)+2:GOSUB 7100
4235 IF S=0 THEN PRINT TAB(XR(S)){M}:GOTO 4249
4240 FOR I=0 TO 4-S:PRINT TAB(XR(S)+I){M}:NEXT
4249 PRINT "{home}"
4250 RETURN
4300 J=3-S:IF S=0 THEN J=0
4305 PRINT "{home}";
4306 IF S>0 GOTO 4328
4307 PRINT TAB(20);:PRINT "{down}";:FOR I=1 TO 18:PRINT "{#}";:NEXT:
PRINT "{#}"
4310 DN=N2(0):GOSUB 7100
4320 PRINT TAB(20);:FOR I=1 TO 19:PRINT "${}";:NEXT:GOTO 4398
4328 DN=YL(S):GOSUB 7100
4329 FOR I=1 TO 5-S:PRINT TAB(J+XR(S)+2){%}:NEXT
4330 PRINT TAB(20);:FOR I=19 TO J+XR(S):PRINT "{#}";:NEXT:
PRINT "{%}"
4370 IF S<4 THEN FOR I=1 TO N2(S):PRINT TAB(J+XR(S)+2){%}:NEXT
4380 PRINT TAB(20);:FOR I=19 TO J+XR(S):PRINT "${}";:NEXT:
PRINT "{%}"
4385 FOR I=1 TO 5-S:PRINT TAB(J+XR(S)+2){%}:NEXT
4398 PRINT "{home}"
4399 RETURN
4400 PRINT "{home}";
4406 IF S>0 GOTO 4430
4407 PRINT "{down}";:FOR I=1 TO 11+XL(0):PRINT "{#}";:NEXT:
PRINT "{#}"
4410 DN=N2(0):GOSUB 7100
4420 FOR I=1 TO 12+XL(0):PRINT "${}";:NEXT:GOTO 4398
4430 DN=YL(S):GOSUB 7100

```

```

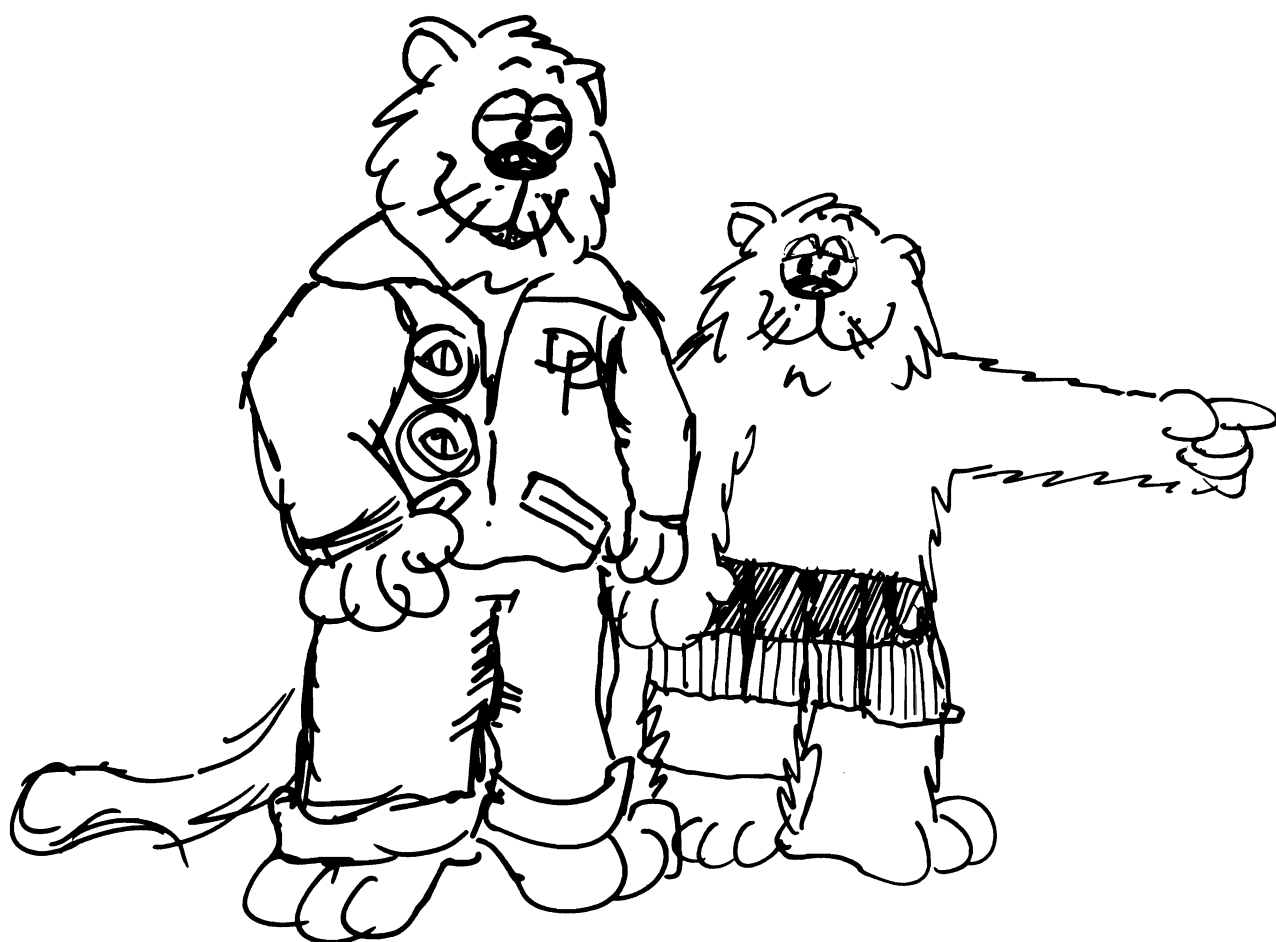
4431 FOR I=1 TO 5-S:PRINT TAB(XL(S))"{'}":NEXT
4432 PRINT TAB(XL(S));:PRINT "{#}";:FOR I=XL(S) TO 17:PRINT "{#}";:
NEXT:PRINT "{#}"
4470 IF S<4 THEN FOR I=1 TO N2(S):PRINT TAB(XL(S))"{'}":NEXT
4480 PRINT TAB(XL(S));:PRINT "{'}";:FOR I=XL(S) TO 17:PRINT "{$}";:
NEXT:PRINT "{$}"
4485 FOR I=1 TO 5-S:PRINT TAB(XL(S))"{'}":NEXT
4498 PRINT "{home}"
4499 RETURN
4500 DN=YL(S):GOSUB 7000
4515 IF S=0 THEN PRINT TAB(XL(S));"{$}"
4520 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XL(S))"{'}":NEXT
4545 IF S>0 THEN PRINT TAB(XL(S))LEFT$("{'$$$$}",6-S)
4565 IF S=4 THEN PRINT TAB(XL(S))"{' ' 2°left down '}"
4570 Q$="":IF S>0 THEN Q$=MID$("{3°right '}",S)
4572 IF S<4 THEN FOR I=0 TO N2(S)+1:PRINT TAB(XL(S))"{'}"Q$:NEXT
4575 IF S>0 THEN PRINT TAB(XL(S))LEFT$("{'####}",6-S)
4595 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XL(S))"{'}":NEXT
4596 IF S=0 THEN PRINT TAB(XL(S))"{#}"
4597 PRINT "{home}"
4599 RETURN
4700 DN=YL(S):GOSUB 7000
4715 IF S=0 THEN PRINT TAB(XL(S));"{M}":GOTO 4730
4720 FOR I=1 TO 5-S:PRINT TAB(XL(S)+I)"{M}":NEXT
4730 DN=N2(S)+2:GOSUB 7100
4735 IF S=0 THEN PRINT TAB(XL(S))"{N}":GOTO 4749
4740 FOR I=0 TO 4-S:PRINT TAB(XL(S)+5-S-I)"{N}":NEXT
4749 PRINT "{home}"
4750 RETURN
4800 GOSUB 4200
4815 DN=YL(S)+5-S:GOSUB 7000
4820 PRINT TAB(20);
4825 IF S<4 THEN FOR I=1 TO XR(S)-21:PRINT "{#}";:NEXT
4830 PRINT "{P left down}";
4835 IF S<4 THEN FOR I=1 TO N2(S):PRINT "{ ' down left}";:NEXT
4840 PRINT "{' : 2°left}";
4850 IF S<4 THEN FOR I=1 TO XR(S)-20-1:PRINT "{$ 2°left}";:NEXT
4851 PRINT "{home}"
4855 RETURN
4900 IF S=0 THEN J=5:GOTO 4910
4902 J=0
4910 GOSUB 4700
4915 DN=YL(S)+5-S:GOSUB 7000
4920 PRINT TAB(19);
4925 IF S<4 THEN FOR I=1 TO 18-XL(S)-5+S+J:PRINT "{# 2°left}";:NEXT
4930 PRINT "{O left down}";
4935 IF S<4 THEN FOR I=1 TO N2(S):PRINT "{% down left}";:NEXT
4940 PRINT "{L}";
4950 IF S<4 THEN FOR I=1 TO 18-XL(S)-5+S+J:PRINT "{$}";:NEXT
4951 PRINT "{home}"
4955 RETURN

```

```

5000 IF S=0 OR S=4 THEN RETURN
5010 DN=18:GOSUB 7000
5020 IF S=3 THEN PRINT TAB(20);"{3°up}."
5030 IF S=2 THEN PRINT TAB(19);"{rvs up ) off ) 2°left down &N up
      %}"
5040 IF S=1 THEN PRINT TAB(18);"{down}_{$$$ 4°left down N}__{N%
      5°left down &&&}__{% 5°left down &&&N}"
5050 RETURN
5100 IF S=0 OR S=4 THEN RETURN
5110 DN=20:GOSUB 7000
5120 IF S=3 THEN PRINT TAB(19);"{5°up rvs "" up 2°left off ""}"
5130 IF S=2 THEN PRINT TAB(16);"{3°up rvs )}_.....{<}"
5141 IF S=1 THEN PRINT TAB(13);"{rvs )}_.....{<}"
5142 IF S=1 THEN PRINT TAB(12);"{rvs )}_.....{<}"
5150 RETURN
6000 K=TI:FOR I=1 TO 25:POKE 245,22:PRINT :
      PRINT TAB(15)"DELICIOUS!":FOR J=1 TO 50:NEXT J
6010 POKE 245,22:PRINT :PRINT TAB(15)"_.....":FOR J=1 TO 50:
      NEXT J,I
6100 GOSUB 2000
6110 PRINT TAB(10);"{up}ANOTHER?_";
6115 SW=1:TM=0
6120 GET T$:IF T$<>" " THEN 6200
6130 IF TI>TM THEN PRINT MID$("?_",SW,1);"{left}";:TM=TI+15:SW=3-SW
6140 GOTO 6120
6200 IF T$="Y" OR T$=CR$ THEN PRINT "YES":GOTO 200
6210 IF T$="N" THEN PRINT "NO{down}":END
6220 GOTO 6120
7000 PRINT "{home}";
7100 IF DN>0 THEN FOR Z=1 TO DN:PRINT "{down}";:NEXT
7110 RETURN
62000 PRINT "{clr down}";TAB(15-LEN(PG$));PG$;"_BY_C._T._NADOVICH"
62010 PRINT "{down}_.....CURSOR_#13_..COPYRIGHT_(C)_1981"
62020 FOR I=1 TO 13:PRINT "{@@@}";:NEXT:PRINT
62030 PRINT "{down}FIND_YOUR_WAY_THROUGH_A_PERSPECTIVE_MAZE"
62032 PRINT "{down}YOUR_CONTROLS_ARE:"
62034 PRINT "{down}_..8_..MOVE_FORWARD"
62035 PRINT "{down}_..4_..TURN_LEFT"
62036 PRINT "{down}_..6_..TURN_RIGHT"
62038 PRINT "{down}_..2_..TURN_AROUND"
62040 PRINT "{down}_..?_..HELP!"
62049 PRINT "{3°down}PRESS_{rvs}RETURN{off}_..TO_BEGIN"
62050 GET T$:IF T$="" THEN 62050
62060 CLR:QK=525:CR$=CHR$(13):IF PEEK(50000) THEN QK=158
62070 PRINT "{clr}":GOTO 100

```

GAMES OF RISK



MAZE

By Howard Arrington

So you want to be rich, do you? Would you be satisfied with numerous 24-carat gold bars? If so, this game is for you.

In MAZE, the PET constructs a maze of passages and rooms, and at your command, fills the rooms with gold. It doesn't do this secretly, either. You're free to watch if you like. When all the gold is stowed away, you get to venture in after it.

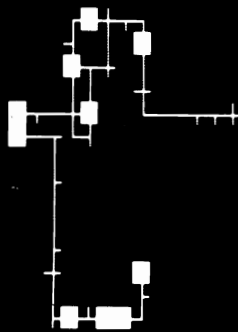
There is one minor hitch. When all the gold is safely in place, the map you were studying vanishes. You will move about by using the numeric keypad (8 moves up, 4 left, 6 right, and 2 down). As you travel, those parts of the maze

that you visit will become visible again. When you enter a room with gold in it, the value of the gold will be announced, and the gold added to your hoard. Beware, if a room has several caches of gold in it, you'll only find one at a time. You must leave the room and re-enter it to find another.

You may wander as long as you like, even after you've found all the gold. To stop your travels, press [RETURN]. If you despair of finding the gold, press [?], and the PET will show you where it all is (but won't let you in after it). One more thing — you're being timed, so don't waste time admiring your newly found riches.

■ TREASURE # 4

HIDDEN 122239 GOLD PIECES
WEALTH 17819 GOLD PIECES



```
0 CLR:POKE 59468,12
6 REM MAZE COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="MAZE":NM$="8":GOSUB 62000
100 GOSUB 900:Q=Q-800:GOSUB 810
110 CLR:GOSUB 60400:GOSUB 900:Q1=WD*20:Q2=CRT+WD*5:
    GP$="GOLD_PIECES"
200 SS=50:DIM A$(SS),C(12),D$(12,4),R$(8,4),B$(4,13),W$(4)
220 FOR I=0 TO 12:READ C(I):NEXT
230 DATA 32,64,107,109,110,125,112,113,114,115,91,93,160
250 FOR I=1 TO 4:FOR J=1 TO 13:READ B$(I,J):NEXT:NEXT
```

```

260 DATA 12,12,12,11,11,11,11,11,3,5,2,7,9,10,12,12,12,1,1,1,1,3,6,2,
    7,8,10
280 DATA 12,12,12,1,1,1,1,4,5,7,8,9,10,12,12,12,11,11,11,11,4,6,2,
    8,9,10
400 FOR I=1 TO 12:FOR J=1 TO 4:READ D%(I,J):NEXT:NEXT
410 DATA 0,1,1,0,1,0,1,1,0,0,1,1,1,1,0,0,0,1,0,1,1,0,1,0,0,1,1,1,1,
    1,1,0
420 DATA 1,1,0,1,1,1,1,1,1,1,0,0,1,1,1,1,1
430 FOR I=1 TO 4:READ D(I),W$(I):NEXT
435 DATA 40,SOUTH,-1,WEST,1,EAST,-40,NORTH
436 D(1)=WD:D(4)=-WD
440 BL$="....."
500 S=1+INT(RND(1)*Q1):E=10:P=91:J%=1
510 PRINT "{clr}NOW,CONSTRUCTING,MAZE.":Z=WD*.4:K=0
516 FOR J=Q TO Q+Q1-1:POKE J,0:NEXT
520 L=S:S1=L:A%(1)=L:POKE L+Q,E:POKE L+Q2,P
610 M=INT(RND(1)*4)+1:F=1
620 L=S+D(M):IF L<0 THEN L=L+Q1
626 IF L>=Q1 THEN L=L-Q1
630 P=PEEK(L+Q2):IF P>32 OR (P=32 AND D%(E,M)=0) THEN
    M=(M AND 3)+1:F=F+1:IF F<5 THEN 620
640 IF F=5 THEN 1200
700 F=INT(RND(1)*13)+1:IF F>9 THEN GOSUB 3000
720 E=B%(M,F):POKE L+Q2,C(E):S=L:POKE L+Q,E:IF E=12 THEN Z=Z-1
733 IF Z>0 THEN F=1:GOTO 620
735 IF K=0 THEN PRINT "{home}PRESS,ANY,KEY,TO,STOP,CONSTRUCTION.":
    K=1
737 GET M$:IF M$="" THEN F=1:GOTO 620
740 GOTO 5000
800 Q=PEEK(QM)+256*PEEK(QM+1):RETURN
810 H=INT(Q/256):L=Q-256*H:POKE QM+1,H:POKE QM,L:RETURN
900 QM=134:QK=525:IF PEEK(50000) THEN QM=52:QK=158
910 GOTO 800
1200 S=A%(J%):IF S=0 THEN 5000
1230 P=PEEK(S+Q2):E=PEEK(S+Q):A%(J%)=0:J%=J%-1:GOTO 610
3000 IF J%<SS THEN J%=J%+1:A%(J%)=L
3030 RETURN
5000 PRINT "{home}";BL$:PRINT "{up}NOW,HIDING,THE,GOLD..."
5005 TR=0:W=0:W1=0:FOR I=1 TO 25
5010 L=INT(RND(1)*Q1):IF PEEK(L+Q)<>12 THEN 5010
5020 A%(I)=L:POKE L+Q2,164:L=INT(RND(1)*10000)+1:TR=TR+L:A%(I+25)=L
5030 PRINT "{home 2°down}HIDDEN";TR;TAB(14);GP$:NEXT I
5100 PRINT "{home}PRESS,{rvs}RETURN{off},TO,BEGIN:"
6000 GET M$:IF M$="" THEN 6000
6018 L=S1:PRINT "{clr}":TI$="000000"
6030 E=PEEK(L+Q):P=C(E):P1=P
6040 POKE 32768,P:GET A$
6042 POKE L+Q2,P1
6045 IF A$="?" THEN 7000
6047 IF A$="Q" OR A$=CHR$(13) THEN 7100
6050 IF A$="" THEN P1=(P1+128) AND 255:GOTO 6040

```

```

6060 POKE L+Q2,P
6080 J=ASC(A$)-48:IF J<2 OR J>8 OR (J AND 1) THEN 6040
6100 J%=J/2:J=L+D%(E,J%)*D(J%)
6102 IF J<0 THEN J=J+Q1
6104 IF J>=Q1 THEN J=J-Q1
6110 F=PEEK(J+Q):PRINT "{home 2°right}";W$(J%);"....."
6120 IF D%(F,5-J%)=0 THEN PRINT "{home}";TAB(10);"BLOCKED":GOTO 6040
6130 L=J:IF F=12 THEN GOSUB 8000
6140 GOTO 6030
7000 PRINT "{home}..**_HERE'S_THE_MAZE:..**"
7010 T$=TI$:FOR J=0 TO Q1-1
7015 E=PEEK(J+Q):F=PEEK(J+Q2)
7016 I=128:IF F<>32 THEN I=-58*(E=12)
7017 IF E=12 THEN I=-I
7020 POKE J+Q2,C(E)+I:NEXT J
7030 FOR I=1 TO 25:IF A%(I)=-1 THEN 7070
7040 T=Q2+A%(I):F=PEEK(T):IF F=32 OR F=36 THEN POKE T,36
7050 IF F<>32 AND F<>36 THEN POKE T,164
7070 NEXT I
7080 TI$=T$
7100 T$=TI$:PRINT "{home down}...TIME.";LEFT$(T$,2);": ";MID$(T$,3,
2);": ";RIGHT$(T$,2);
7110 PRINT "....."
7200 PRINT "{home}";BL$:PRINT "{up}WANT_TO_PLAY_AGAIN?.";
7210 GOSUB 50000
7220 IF YN THEN FOR I=1 TO 500:NEXT I
7240 Q=Q+Q1:GOSUB 810:CLR
7250 PRINT "{4°down}"
7260 END
8000 FOR I=1 TO 25:IF A%(I)=L THEN 8100
8020 NEXT I:RETURN
8100 A%(I)=-1:I=A%(I+25):TR=TR-I
8105 W1=W1+1:PRINT "{home 2°right}TREASURE.#";W1
8110 PRINT "FOUND.";I;TAB(14);GP$
8120 IF TR>0 THEN PRINT "HIDDEN";TR;"{left}..{left}";TAB(14);GP$
8125 IF TR<=0 THEN PRINT "YOU_FOUND_ALL_THE_HIDDEN_GOLD!"
8140 W=W+I:PRINT "WEALTH";W;TAB(14);GP$
8150 FOR I=1 TO 500:IF PEEK(QK)<>0 THEN 8160
8155 NEXT I
8160 PRINT "{home down}";BL$
8170 RETURN
50000 YN=1
50010 SW=1:TM=0
50020 GET T$:IF T$<>" " THEN 50100
50030 IF TI>TM THEN PRINT MID$("?_",SW,1);"{left}";:SW=3-SW:TM=TI+30
50040 GOTO 50020
50100 IF T$="Y" THEN YN=1:GOTO 50150
50110 IF T$="N" THEN YN=0:GOTO 50150
50120 IF T$<>CHR$(13) THEN 50030
50150 PRINT MID$("NO_YES",1+3*YN,3):RETURN
60200 IF PN=0 THEN 60250

```

```

60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:"
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN PRINT "{clr}":GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_ _CURSOR_#";NM$;
    "_ _ (C) _1981 _ _ _ _PAGE";PN
60270 GOSUB 60500:RETURN
60400 CRT=32768:WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60415 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;
    "_ _BY_HOWARD_ARRINGTON"
62002 PRINT "{down} _ _ _ _CURSOR_#8 _ _COPYRIGHT_(C) _1981"
62004 GOSUB 60500
62010 PRINT "{down}FIND_THE_HIDDEN_TREASURES"
62020 PRINT "{2°down}WANT_DIRECTIONS?_";:YN=0:GOSUB 50010
62030 IF YN=0 THEN PRINT "{clr}";:CLR:POKE 59468,12:GOTO 100
62300 LP=3:GOSUB 60200
62310 PRINT "THE_OBJECT_OF_THIS_GAME_IS_TO_FIND_ALL"
62320 PRINT "THE_GOLD_HIDDEN_IN_A_MAZE."
62360 PRINT "{down}FIRST,_THE_SCREEN_IS_CLEARED,_AND_CON-"
62370 PRINT "STRUCTION_OF_THE_MAZE_IS_BEGUN._AT_ALL"
62375 PRINT "TIMES,_YOU_CAN_SEE_EXACTLY_HOW_DEVIOUS"
62376 PRINT "THE_MAZE_HAD_BECOME._ _WHEN_THE_MAZE_IS{_}"
62380 PRINT "AS_COMPLEX{_}(OR_AS_SIMPLE)_AS_YOU_LIKE,"
62390 PRINT "PRESS{_rvs}RETURN{off}_{TO_STOP_CONSTRUCTION."
62420 PRINT "{down}WHEN_THE_MAZE_IS_COMPLETE,_THE_PET"
62430 PRINT "HIDES_A_NUMBER_OF_GOLD_PIECES_{rvs}${off})_IN"
62440 PRINT "VARIOUS_PLACES_IN_THE_MAZE."
62475 GOSUB 60200
62480 PRINT "ONCE_ALL_THE_GOLD_IS_HIDDEN,_YOU_MAY"
62490 PRINT "BEGIN_YOUR_SEARCH_BY_PRESSING_{rvs}RETURN{off})."
62500 PRINT "THE_MAZE_WILL_VANISH,_AND_YOU_WILL_FIND"
62510 PRINT "YOURSELF_AT_THE_START_OF_THE_MAZE."
62520 PRINT "YOU_CAN_MOVE_BY_PRESSING_THE{_rvs}8{off},_{rvs}4{off},_ _8"
62530 PRINT "{rvs}6{off},_AND_{rvs}2{off}_KEYS._{rvs}8{off}
    _MOVES_UP,_{rvs}4{off}_LEFT,_ _{}"
62540 PRINT
    "{rvs}6{off}_RIGHT,_AND_{rvs}2{off}_MOVES_DOWN._ _ _WHEN-_4{@[@}6"
62550 PRINT "EVER_YOU_STUMBLE_OVER_SOME_GOLD,_ _ _ _{}"
62560 PRINT "ITS_VALUE_WILL_BE_ADDED_TO_YOUR_ _ _ _2"
62570 PRINT "WEALTH._ _BEWARE,_ _HOWEVER!_SEVERAL"
62580 PRINT "PILES_OF_GOLD_PIECES_MAY_BE_HIDDEN_AT"
62590 PRINT "ANY_GIVEN_SPOT,_AND_YOU_MUST_RE-ENTER"
62600 PRINT "THE_HIDING_PLACE_TO_FIND_THE_OTHERS."
62610 GOSUB 60200
62620 PRINT "IF_YOU_FIND_ALL_THE_GOLD,{_}YOU_WILL_BE"

```

```
62630 PRINT "TOLD.SO..YOU.WILL.BE.FREE.TO.CONTINUE"
62640 PRINT "EXPLORING.THE.MAZE,.HOWEVER..AT.ANY"
62650 PRINT "TIME,.YOU.CAN.STOP.EXPLORING.BY.PRESS-"
62660 PRINT "ING{. rvs}RETURN{off}..IF.YOU.DESPAIR.OF.EVER"
62670 PRINT "FINDING.ALL.THE.GOLD,.PRESS.{rvs}?{off},.AND.YOU"
62680 PRINT "WILL.BE.SHOWN.THE.MAZE{.}(AND.THE.GOLD)"
62690 PRINT "AGAIN."
62750 PRINT TAB(11)"_{3°down rvs}HAPPY.HUNTING!{off 2°down}"
62760 GOSUB 60200
```




DUNGEON

By Brian Sawyer

The evil magician Trent has transported you into the depths of some long-abandoned dungeons. The dungeons have been taken over by a large collection of monsters, each with its own hoard of gold. Trent has agreed to release you from his power if (and only if) you somehow get *all* the gold away from the monsters.

The dungeons themselves are the usual run-of-the-mill dungeons — dank, dark, dismal, and decrepit. These dungeons have a novelty, however. Being long-abandoned, several passages have caved in, blocking the route to the rooms they once led into. Trent has foreseen this, and has given you a way to pass through walls. Also, the various creatures have been there a long time (feeding on those unwise enough to come in seeking gold) and no longer fit through the doorways. Finally, the spell Trent used to put you there won't permit you to leave. Only *he* can get you out.

On the screen, you'll see a map of the dungeon, drawn as you travel about (your movement is controlled by the numeric keypad). Your position is marked with a dot. Gold, as you find it, is marked with a "G." The various sorts of creatures are marked with various sorts of markers. Open space is shown as white, and doors as grey squares. Walls are shown as black.

Again, nothing will show up on the screen until you find it in the dungeon.

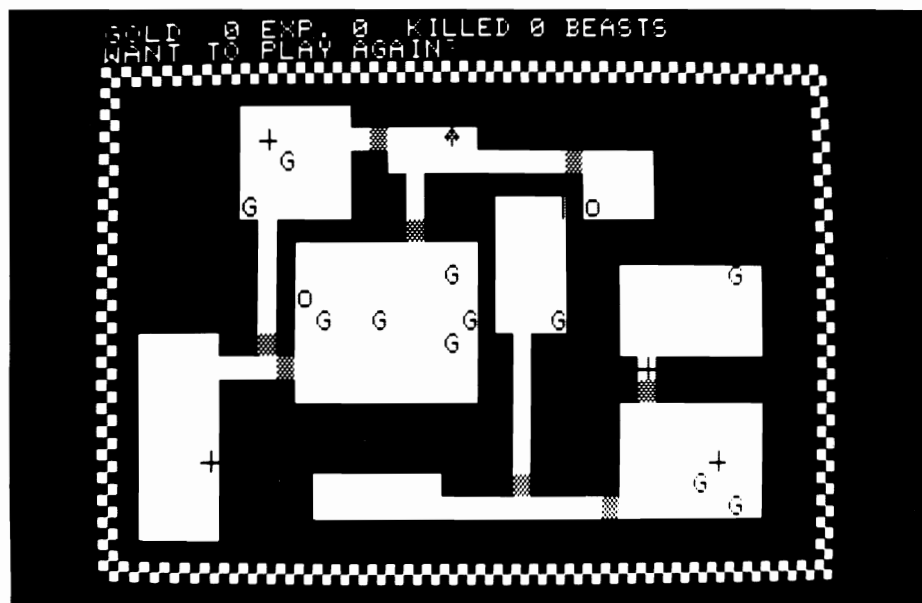
In addition to the map, lines of information are kept on the screen. They show your hit points, your experience points, and how much gold you have.

Hit points measure how much strength you have (zero means you're dead). Wandering about doing nothing uses up a few hit points; battling a monster uses up a great many. You may rest if your hit points get too low, but do so in a safe place. The monsters find resting humans to be just as tasty as any other kind.

Experience points indicate how good you are at surviving. The more creatures you defeat, the higher your experience points go. The more experience points you get, the easier it is to defeat creatures of a given strength.

To move, use the numeric pad in the usual fashion. To pick up gold, move on top of it. To attack a creature, move on top of it. If you want to move through the wall, press [SHIFT] with the appropriate number. (Walking through walls uses up more energy, and more hit points, than walking through open space.) If you want to quit (and suffer whatever *other* torments Trent can devise), press [Q].

As the Boy Scouts say, be prepared.



```

0 CLR:POKE 59468,12
6 REM DUNGEON COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81
90 PG$="DUNGEON":NM$="15":GOSUB 62000
100 PRINT CHR$(142):RS=23:CS=WD:SZ=RS*CS:BL=(25-RS)*CS:RS=RS-1:
    CS=CS-1
110 DIM DG((SZ+BL)/5),DR(9):REM DG() USED TO GRAB MEMORY
115 P=WD:FOR I=2 TO 8 STEP 3:FOR N=-1 TO 1:DR(I+N)=P+N:NEXT N:
    P=P-WD:NEXT I
120 ER$="{home down}.....{home
    down}"
130 E2$="{home}.....{home}"
140 A=0:C0=0:CC=0:E1=0:GN=0:I=0:IU=0:L=0:N=0:N1=0:O1=0:P=0:PW=0:
    QQ=0:R0=0
150 Q=0:SX=0:TS=0:U=0:V=0:V1=0:Y=0:ZC=0:ZT=0
160 EX$="EXP.":HP$="HIT_PTS.":IF WD>40 THEN EX$="EXPERIENCE":
    HP$="HIT_POINTS"
170 L$="":E$=""
180 PRINT "{clr}SETTING UP..."
190 HP=50:MG=0:EX=0:PX=0:HG=0:Z=0:FG=0:K1=0:E=0:S=0:W=160:ET=160
200 TI$="000000":TM=TI+3600*WD/40
210 AX=CRT:TS=PEEK(QA)+256*PEEK(QA+1)+7+BL
220 FOR I=TS+WD TO TS+SZ-WD:POKE I,32:NEXT I
230 GOSUB 380
240 TS=TS-BL
250 L=INT(RND(1)*SZ+TS):IF PEEK(L)<>160 THEN 250
260 TM=0:GOSUB 1410:L=L+AX-TS:W=PEEK(L):GOSUB 600:POKE L,209:W=160
270 IF PEEK(TS+BL)<>127 THEN PRINT "{home rvs}ABORT!!":GOTO 1360
275 POKE QK,0:POKE QP,255:GOSUB 1240
280 HP=HP-.15-2*SX:IF HP<0 THEN 1190
300 IF (PEEK(L+A)=32) AND (SX<>1) THEN 270
310 IF PEEK(L+A)=127 THEN 270
320 POKE L,W:L=L+A:W=PEEK(L):POKE L,209:GOSUB 600:POKE L,209
330 IF W=135 THEN GOSUB 1200
340 IF W>=214 AND W<=219 THEN GOSUB 1000
350 IF E>0 THEN S=S+1
360 IF S>1 THEN GOSUB 830
370 GOTO 270
380 PRINT INT((TM-TI)/60);"{left}..{up}"
400 W=INT(RND(1)*9+2):L=INT(RND(1)*9+2)
410 R0=INT(RND(1)*(RS-L-1))+1:C0=INT(RND(1)*(CS-W-1))+1:
    P=TS+WD*R0+C0
420 IF P+WD*L+W>=TS+SZ THEN 530
430 FOR N=0 TO L+1:FOR N1=0 TO W+1:IF PEEK(P+(N*WD)+N1)<>32 THEN
    530
440 NEXT N1,N
450 FOR N=1 TO L:FOR N1=1 TO W:POKE P+(N*WD)+N1,160:NEXT N1,N
460 FOR N=P+2+(L+1)*WD TO TS+SZ-1 STEP WD
465 Q=PEEK(N):POKE N,0
470 IF Q=160 THEN 485
475 FOR A=1 TO 100:NEXT A

```